Cagri Ozcinar, PhD

LTTI.00.032 – Machine Learning in Synthetic Biology

Machine Learning

≈Looking for function

Cagri Ozcinar, PhD

Machine Learning

≈Looking for function

- Speech Recognition $f($  $)=$ "How are you"

Machine Learning

≈Looking for function

- Speech Recognition $f($  $)=$ "How are you"
- Image Recognition $f($  $)=$ "Cat"

Cagri Ozcinar, PhD

Machine Learning

≈Looking for function

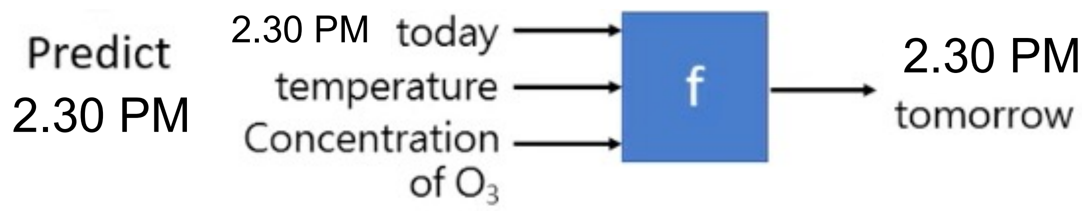
- Speech Recognition $f($  $)=$ "How are you"
- Image Recognition $f($  $)=$ "Cat"
- Playing Go $f($  $)=$ "5-5"

Different types of Functions

- Regression: The function outputs a scalar.

Different types of Functions

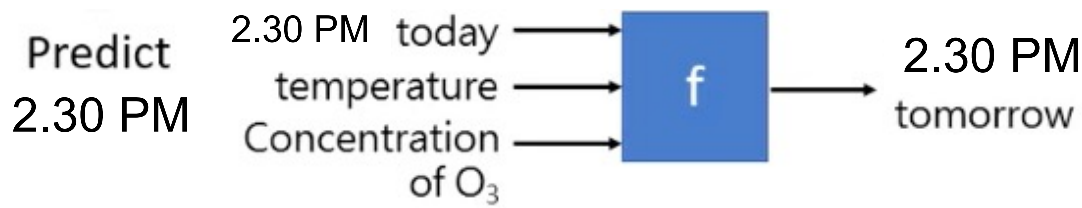
- Regression: The function outputs a scalar.



Cagri Ozcinar, PhD

Different types of Functions

- Regression: The function outputs a scalar.

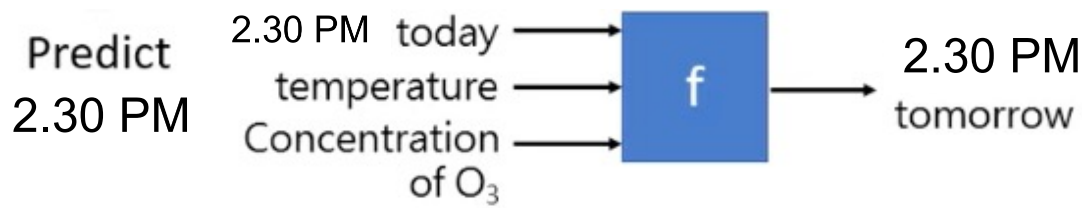


- Classification: Given options (classes), the function outputs the correct one.

Cagri Ozcinar, PhD

Different types of Functions

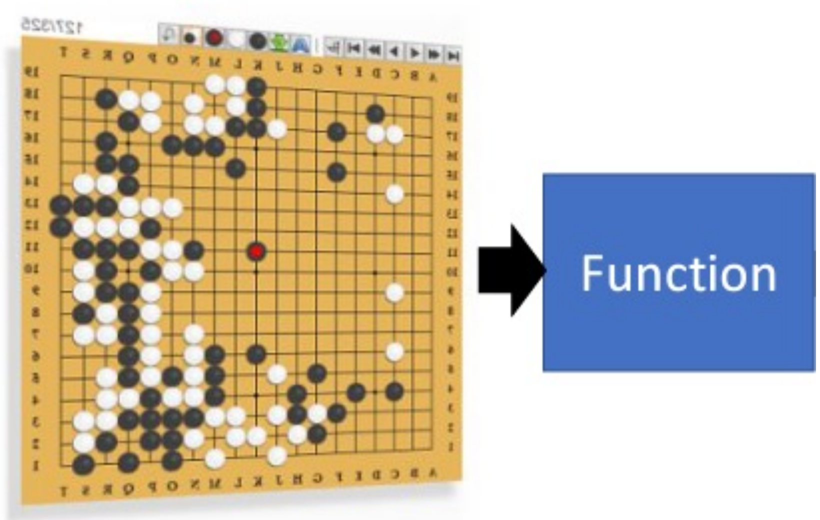
- Regression: The function outputs a scalar.



- Classification: Given options (classes), the function outputs the correct one.

Different types of Functions

- Classification: Given options (classes), the function outputs the correct one.

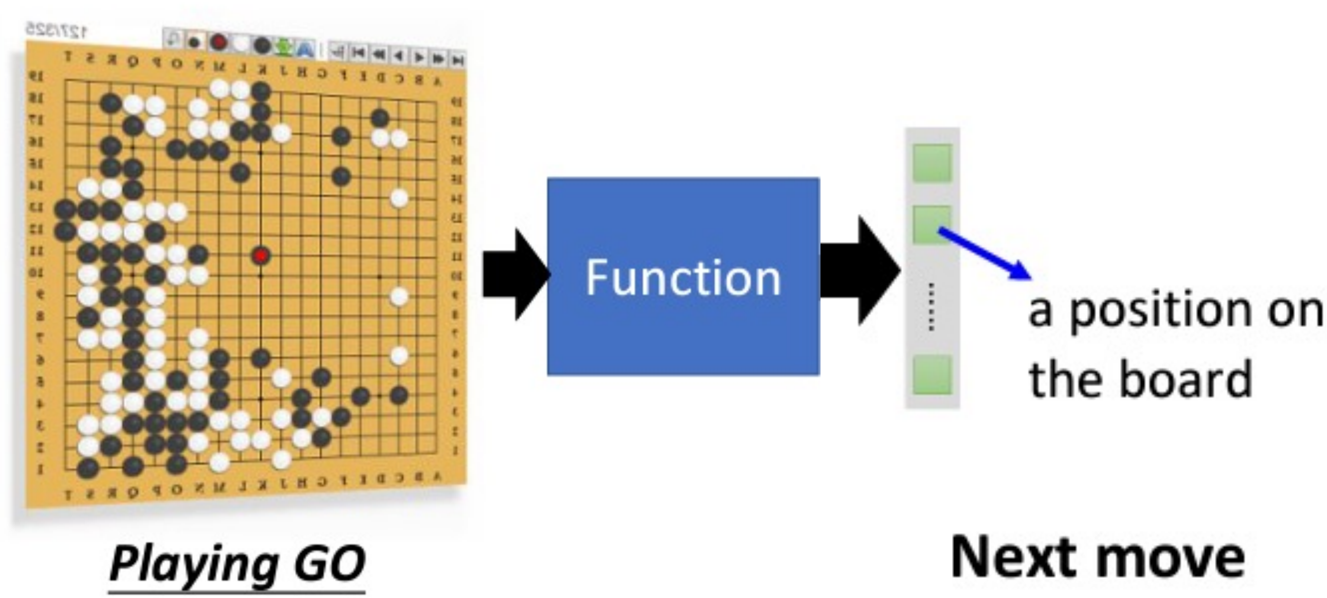


Playing GO

Cagri Ozcinar, PhD

Different types of Functions

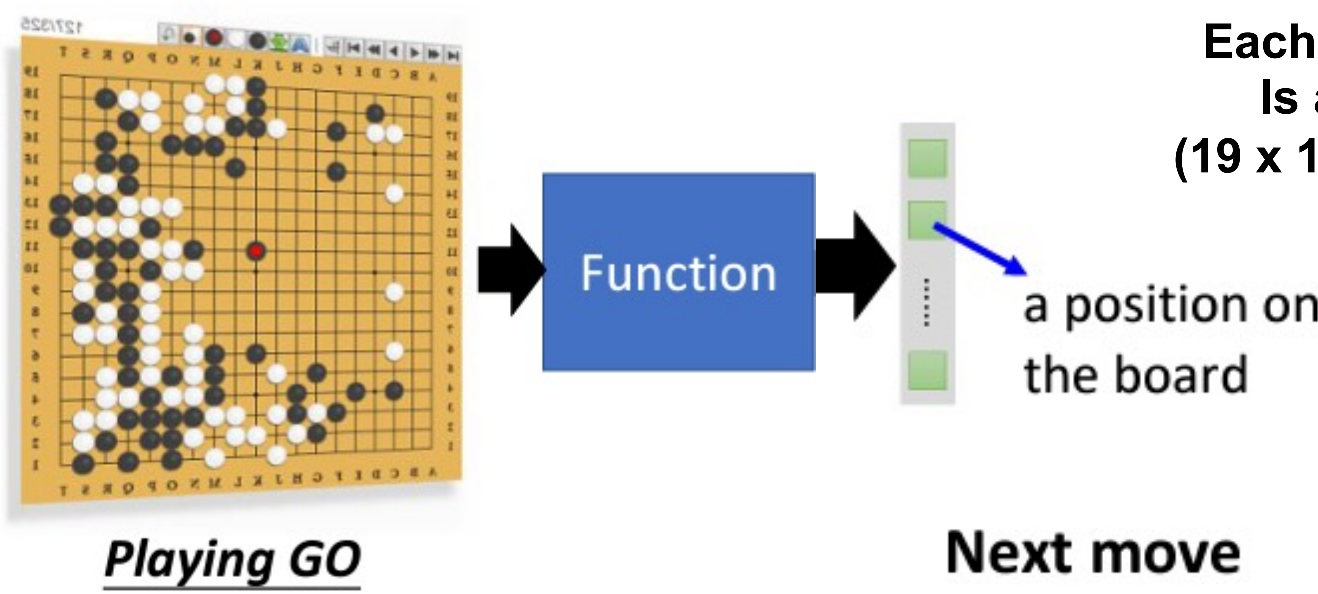
- Classification: Given options (classes), the function outputs the correct one.



Different types of Functions

- Classification: Given options (classes), the function outputs the correct one.

Each position Is a class (19 x 19 Classes)



Structured learning

- Create something with structure (image, document, sound)

Framework of ML

Training data: $\{(\boldsymbol{x}^{\mathbf{1}}, \hat{y}^1), (\boldsymbol{x}^{\mathbf{2}}, \hat{y}^2), \dots, (\boldsymbol{x}^{\boldsymbol{N}}, \hat{y}^N)\}$

Testing data: $\{\boldsymbol{x}^{\boldsymbol{N+1}}, \boldsymbol{x}^{\boldsymbol{N+2}}, \dots, \boldsymbol{x}^{\boldsymbol{N+M}}\}$

Framework of ML

Training data: $\{(\boldsymbol{x}^1, \hat{y}^1), (\boldsymbol{x}^2, \hat{y}^2), \ldots, (\boldsymbol{x}^N, \hat{y}^N)\}$

Testing data: $\{\boldsymbol{x}^{N+1}, \boldsymbol{x}^{N+2}, \ldots, \boldsymbol{x}^{N+M}\}$

Speech Recognition

$\boldsymbol{x}$:  $\hat{y}$: phoneme

Image Recognition

$\boldsymbol{x}$:  $\hat{y}$: soup

Speaker Recognition

$\boldsymbol{x}$:  $\hat{y}$: John (speaker)

Machine Translation

$\boldsymbol{x}$: 痛みを知れ

$\hat{y}$: 了解痛苦吧

Framework of ML

Training data: $\{(\boldsymbol{x}^1, \hat{y}^1), (\boldsymbol{x}^2, \hat{y}^2), \dots, (\boldsymbol{x}^N, \hat{y}^N)\}$

Testing data: $\{\boldsymbol{x}^{N+1}, \boldsymbol{x}^{N+2}, \dots, \boldsymbol{x}^{N+M}\}$

Use $y = f_{\boldsymbol{\theta}^*}(\boldsymbol{x})$ to label the testing data

Framework of ML

Training data: $\{(\boldsymbol{x}^1, \hat{y}^1), (\boldsymbol{x}^2, \hat{y}^2), \ldots, (\boldsymbol{x}^N, \hat{y}^N)\}$

Training:

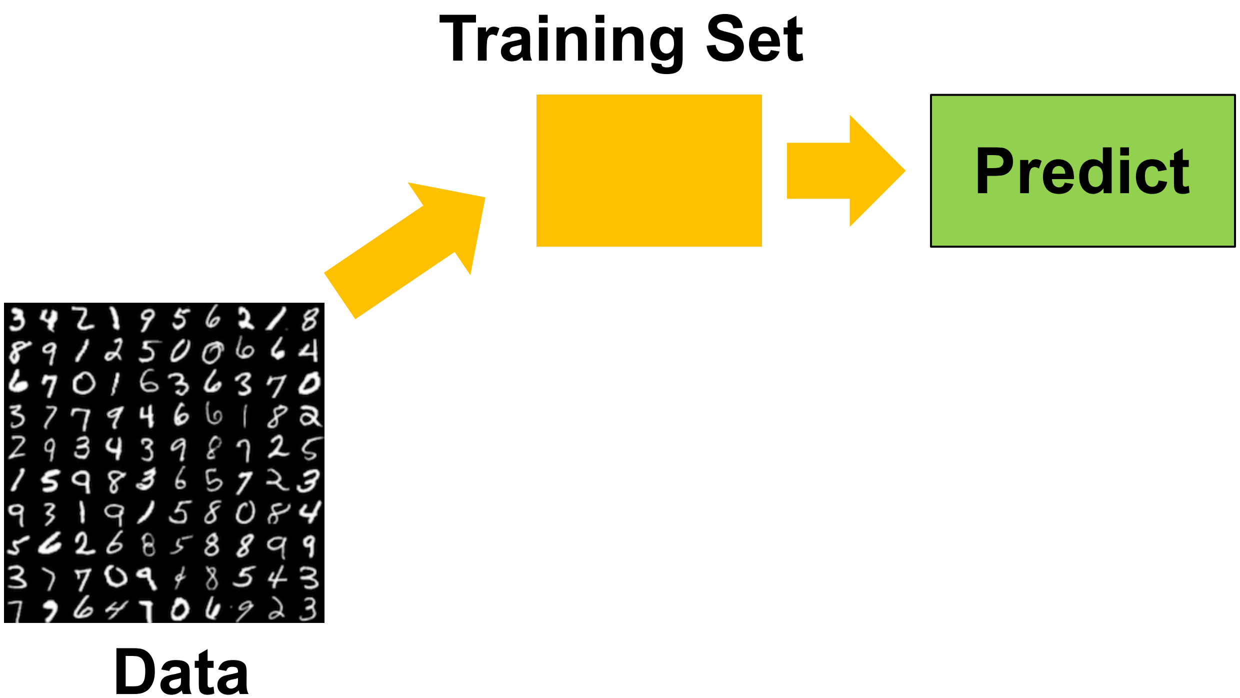
Step 1: function with unknown → **Step 2: define loss from training data** → **Step 3: optimization**

$y = f_{\boldsymbol{\theta}}(\boldsymbol{x})$ $\qquad$ $L(\boldsymbol{\theta})$ $\qquad$ $\boldsymbol{\theta}^* = arg \min_{\boldsymbol{\theta}} L$

Testing data: $\{\boldsymbol{x}^{N+1}, \boldsymbol{x}^{N+2}, \ldots, \boldsymbol{x}^{N+M}\}$

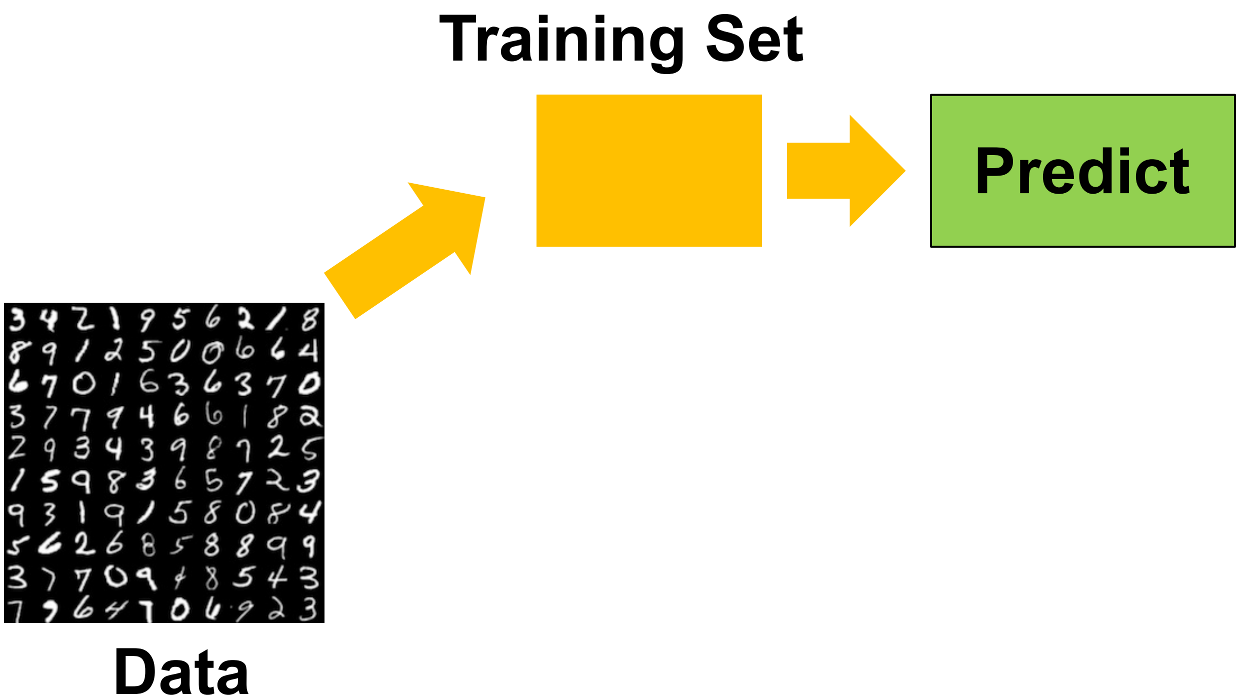
Use $y = f_{\boldsymbol{\theta}^*}(\boldsymbol{x})$ to label the testing data

Basics of deep learning



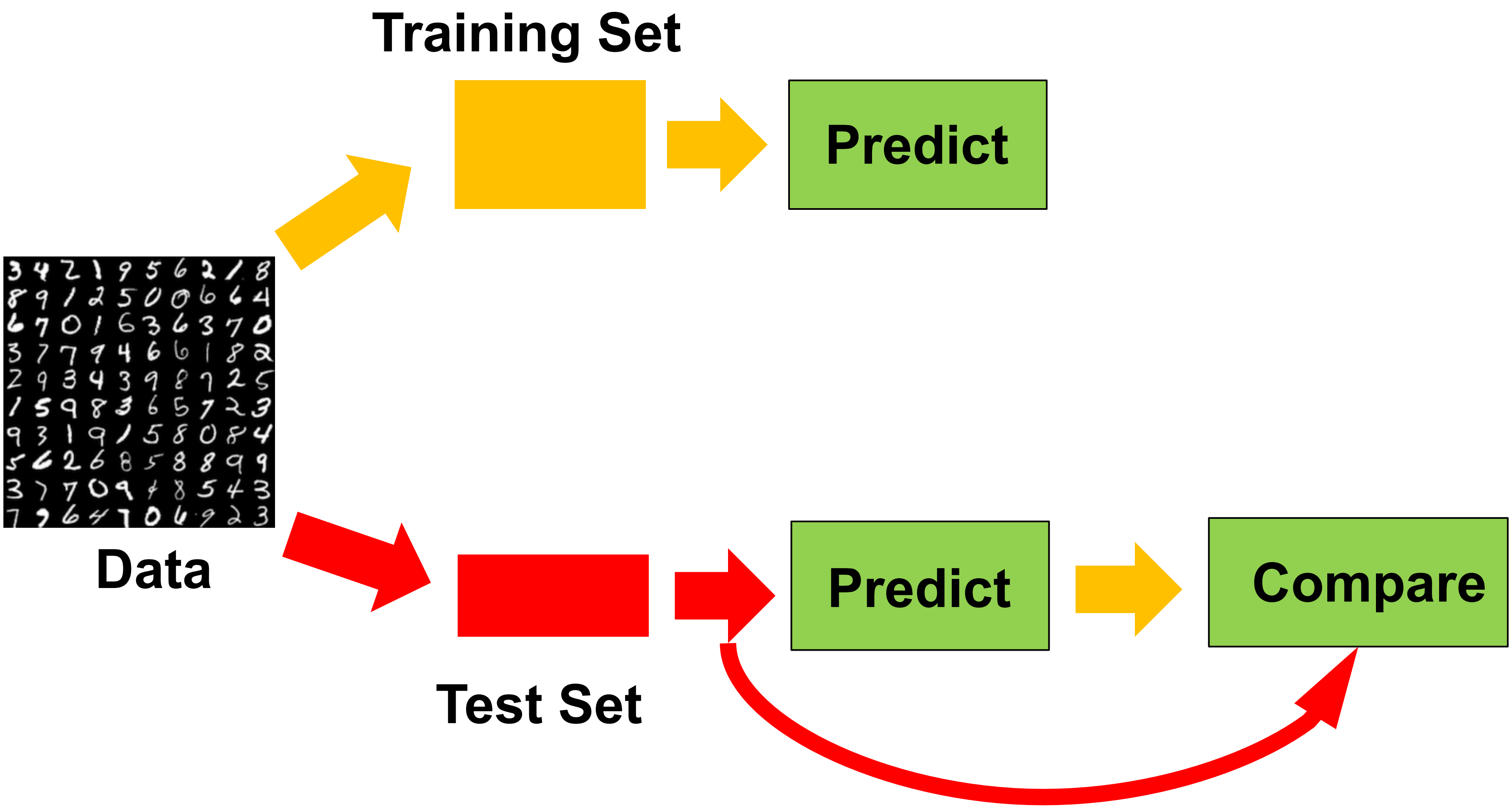
Cagri Ozcinar, PhD

Basics of deep learning



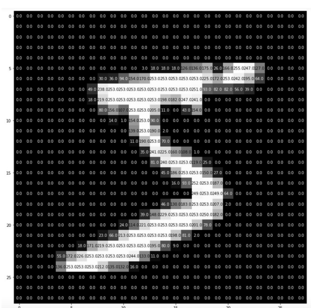
The number of epochs defines the number of times that the learning algorithm will work through the entire training dataset.

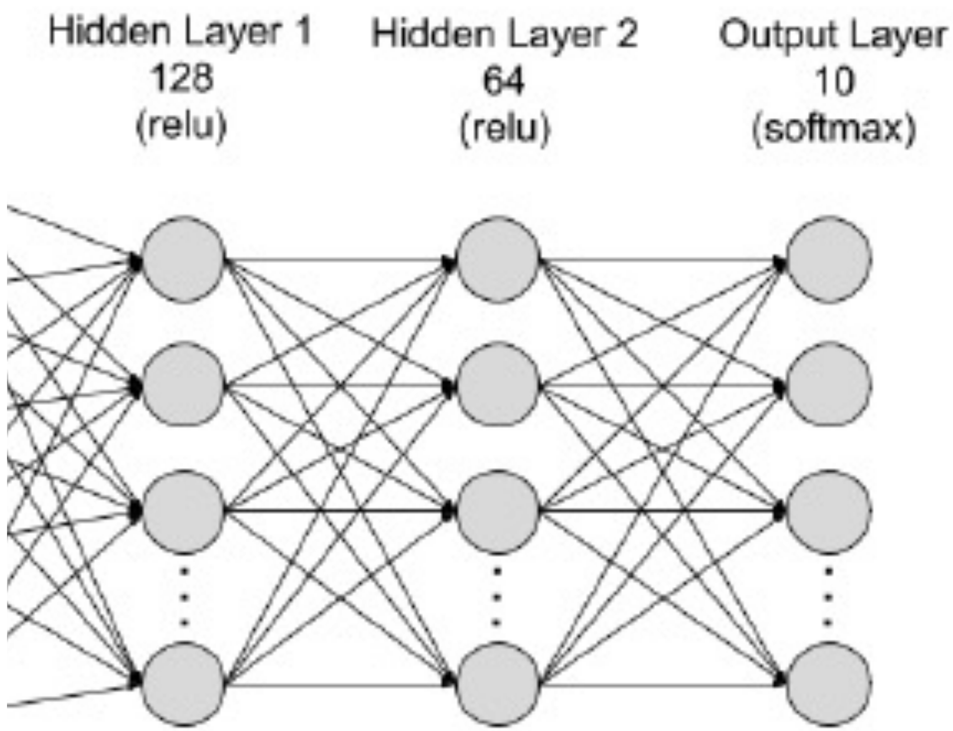
Basics of deep learning



An example neural network

f





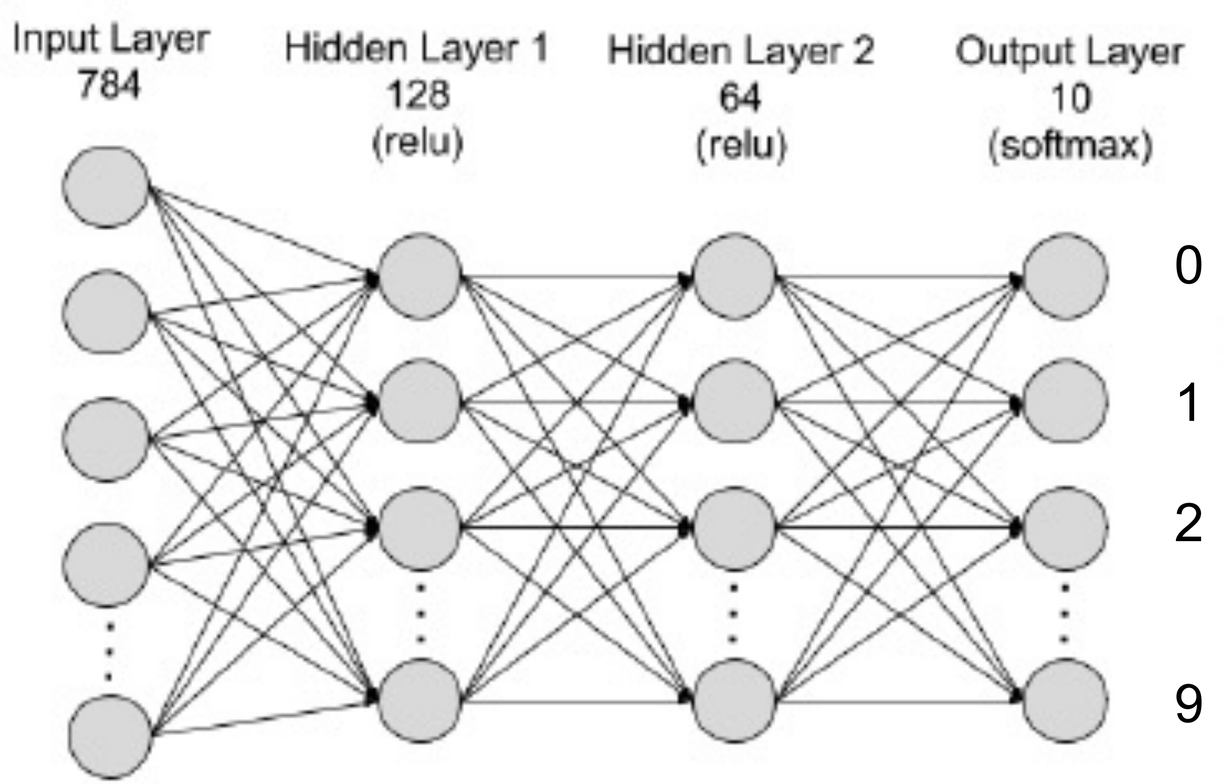
[0,…**5**, ..9]

X: Input: 784 values (28x28)

Y= Output: 10 values

Cagri Ozcinar, PhD

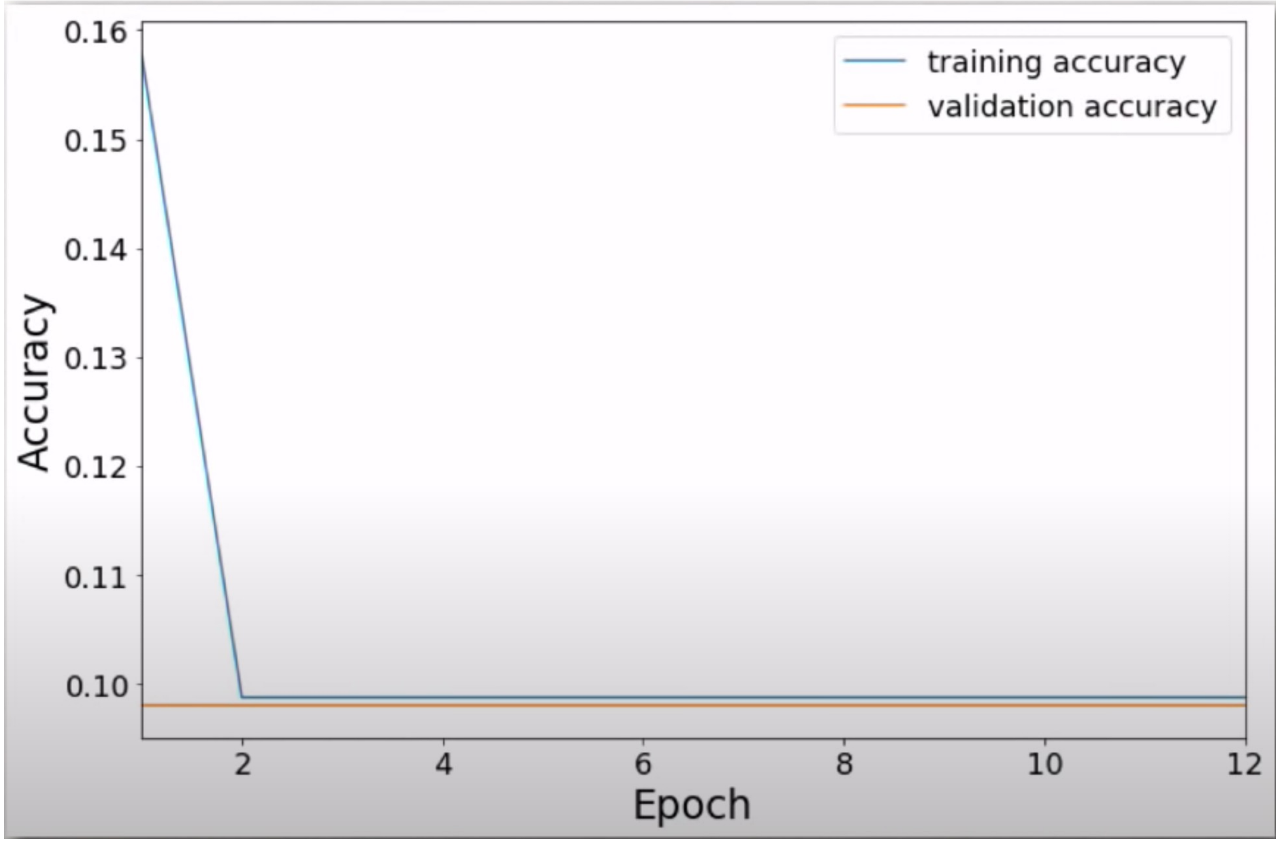
An example neural network



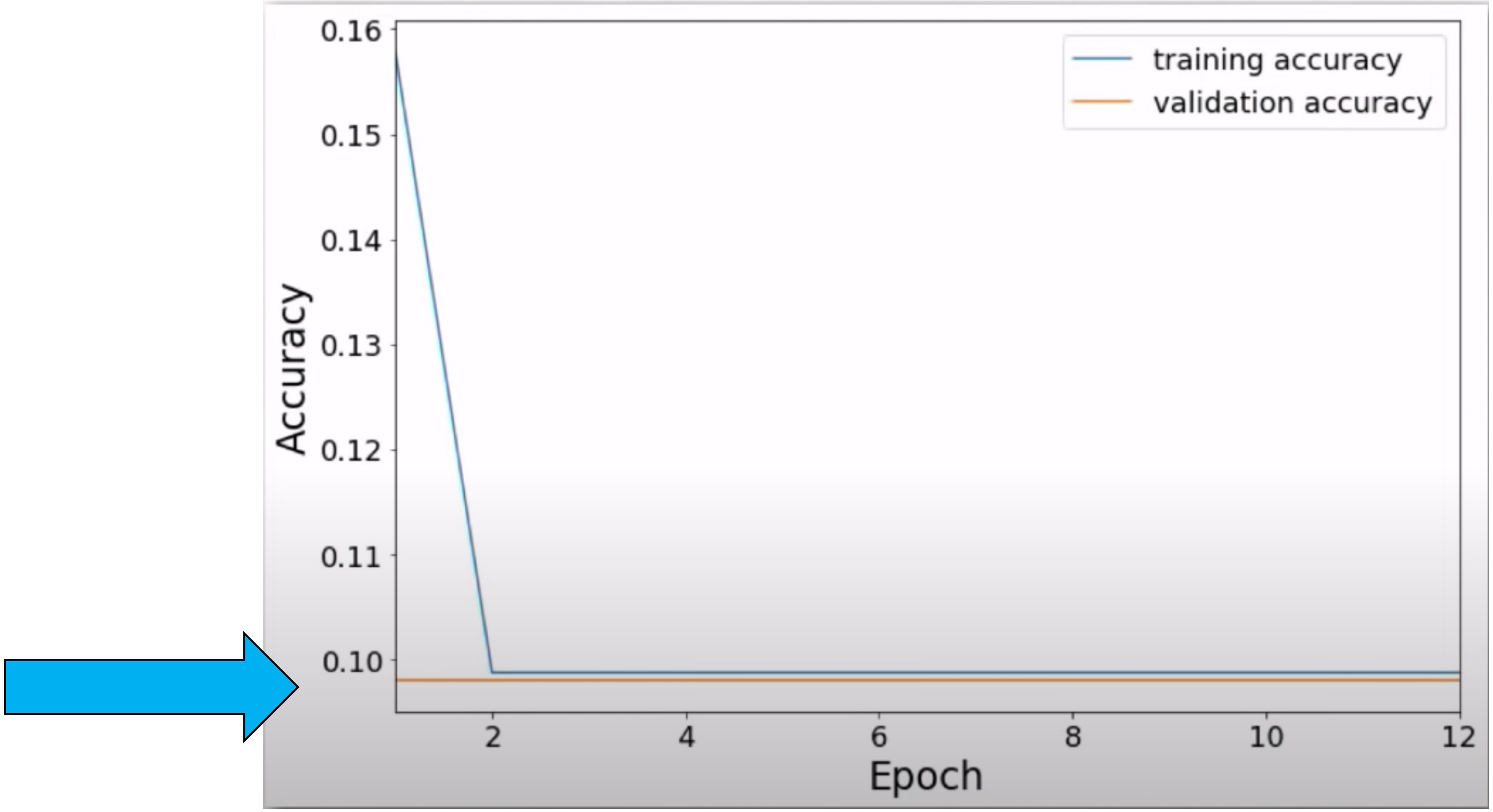
Input: 784 values (28x28)

Output: 10 values

Training Results: Terrible!

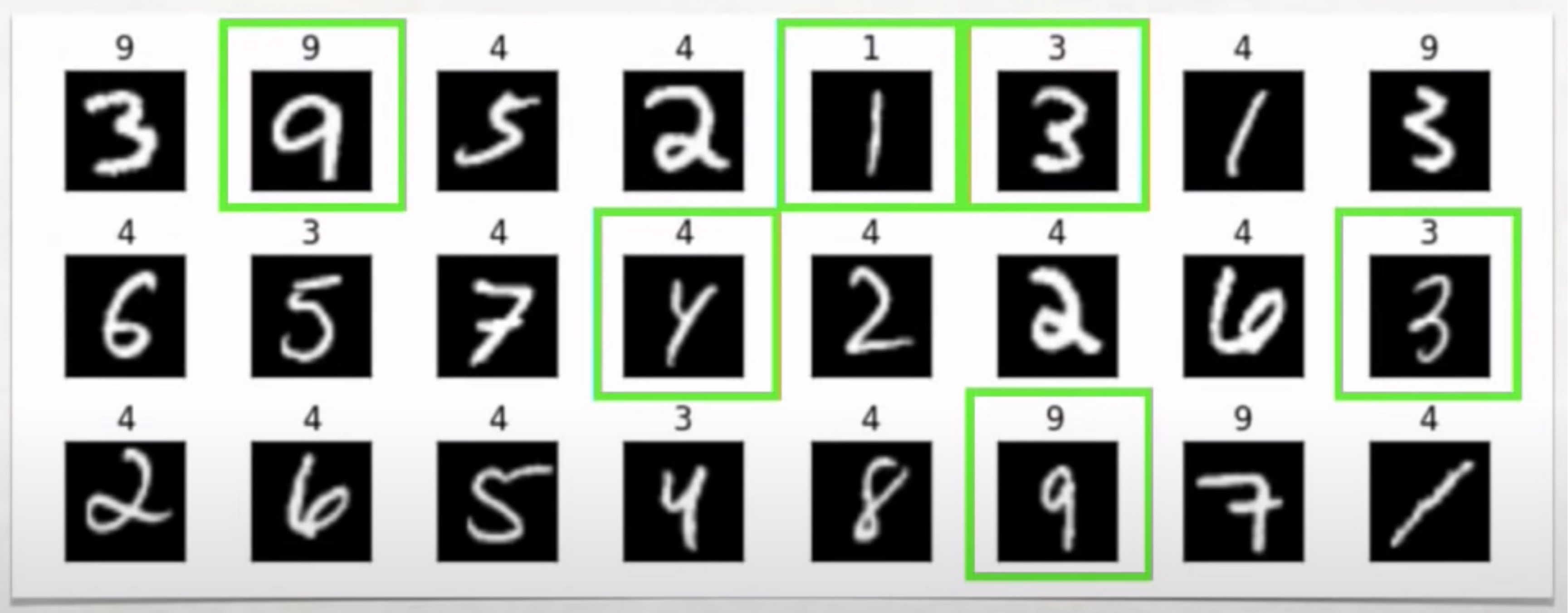


Training Results: Terrible!

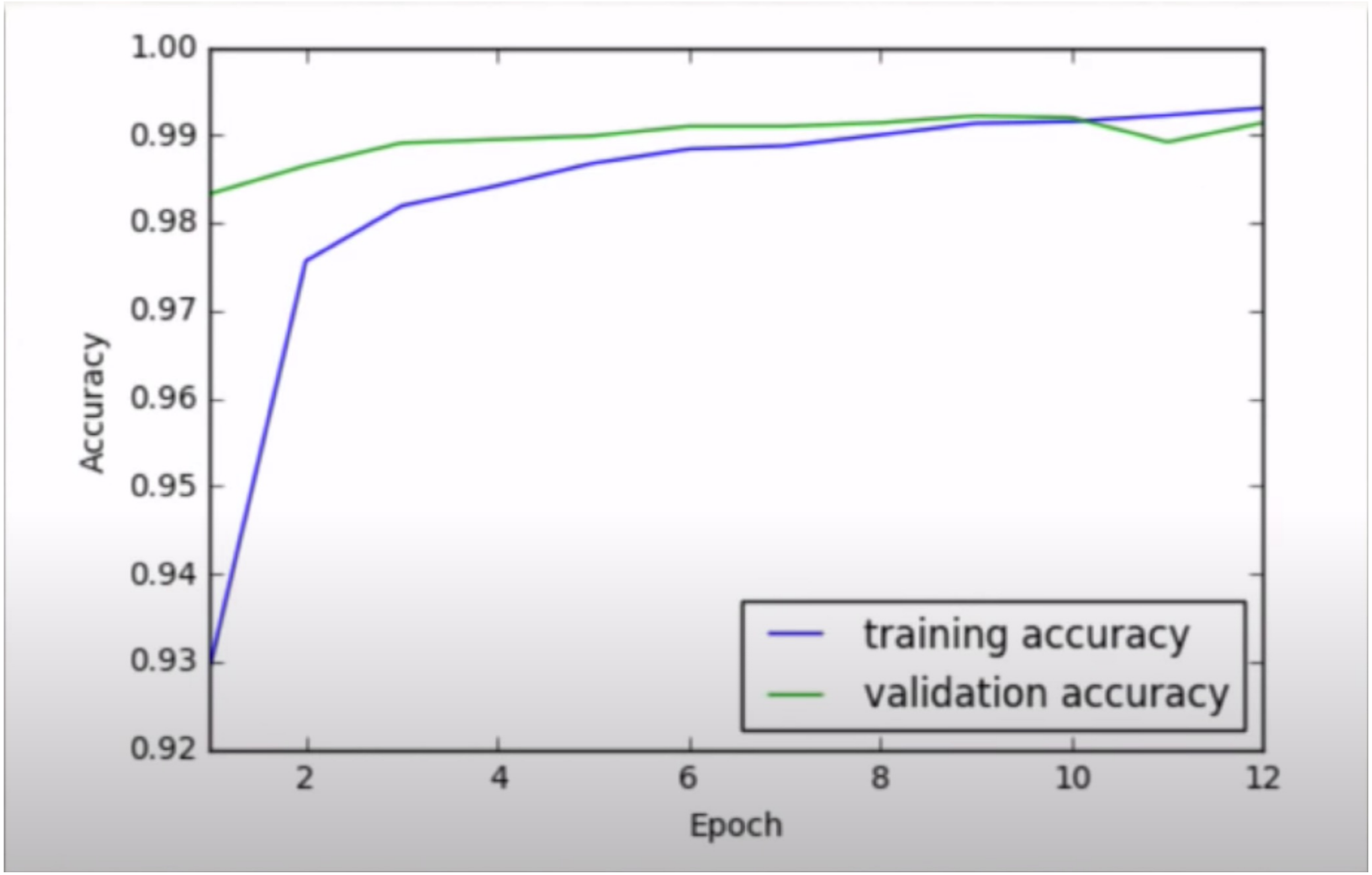


Cagri Ozcinar, PhD

Network Prediction: Terrible!

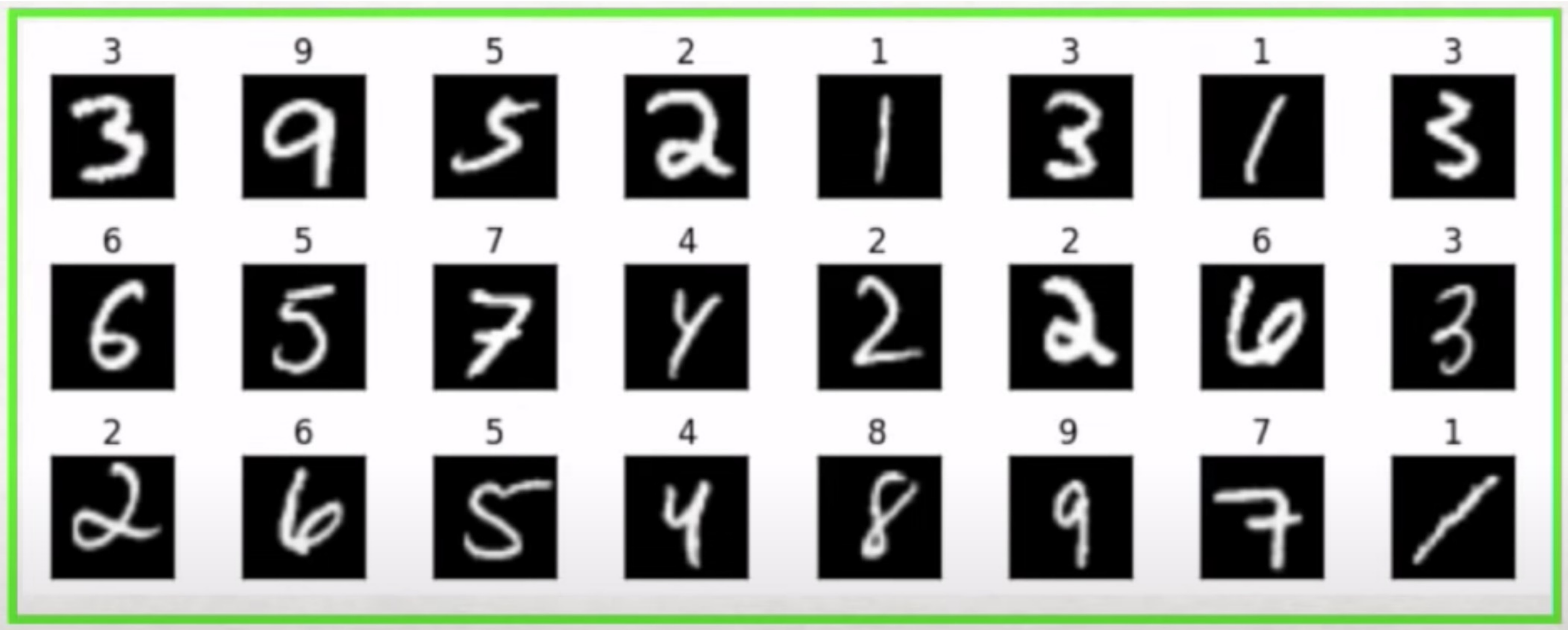


Training Results We Want



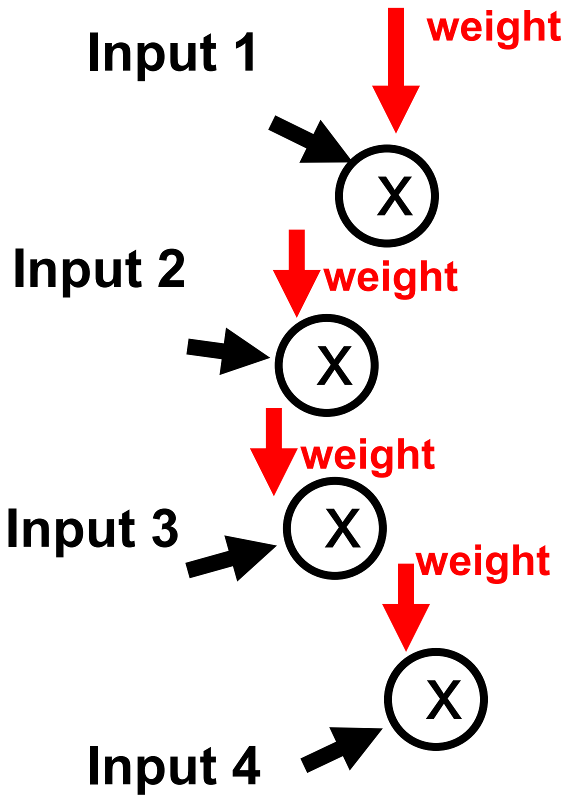
Cagri Ozcinar, PhD

Network Predictions We Want



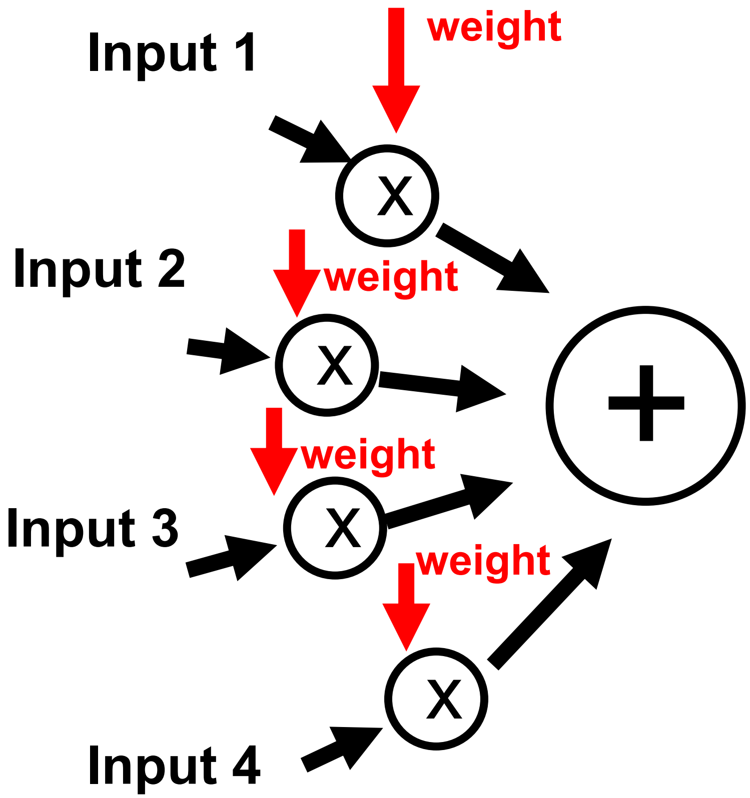
How to
get there

Cagri Ozcinar, PhD

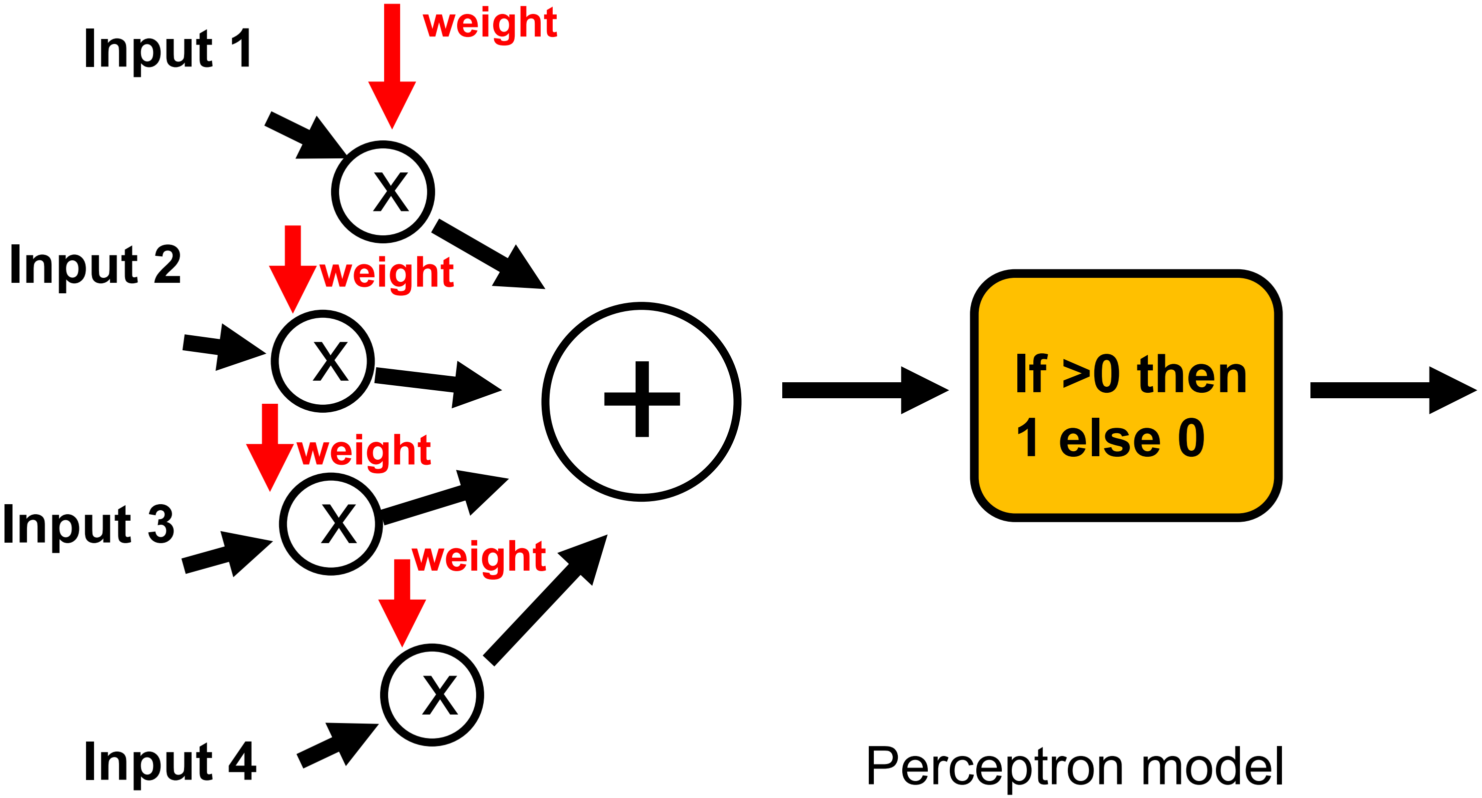


Each neuron has its own weight

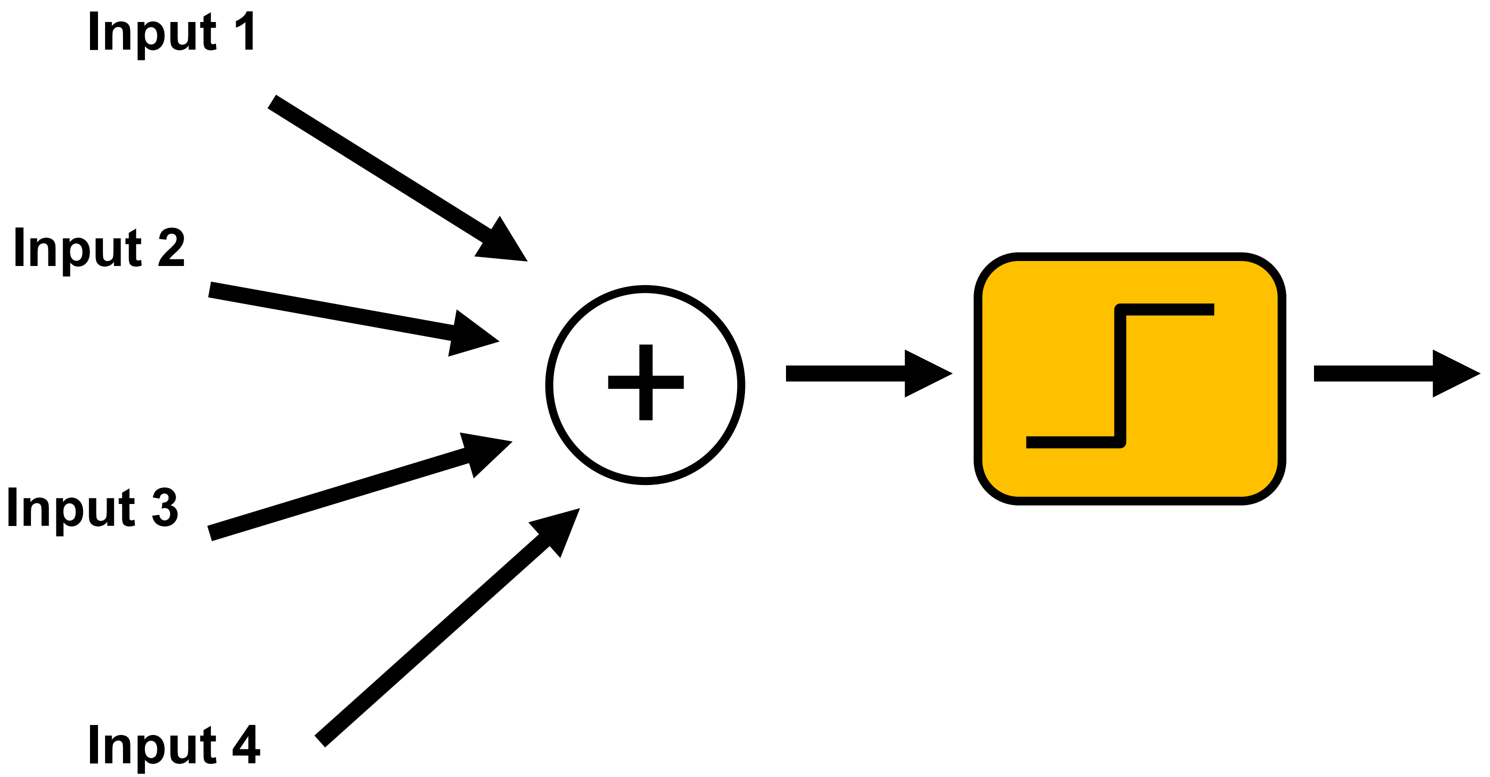
Cagri Ozcinar, PhD



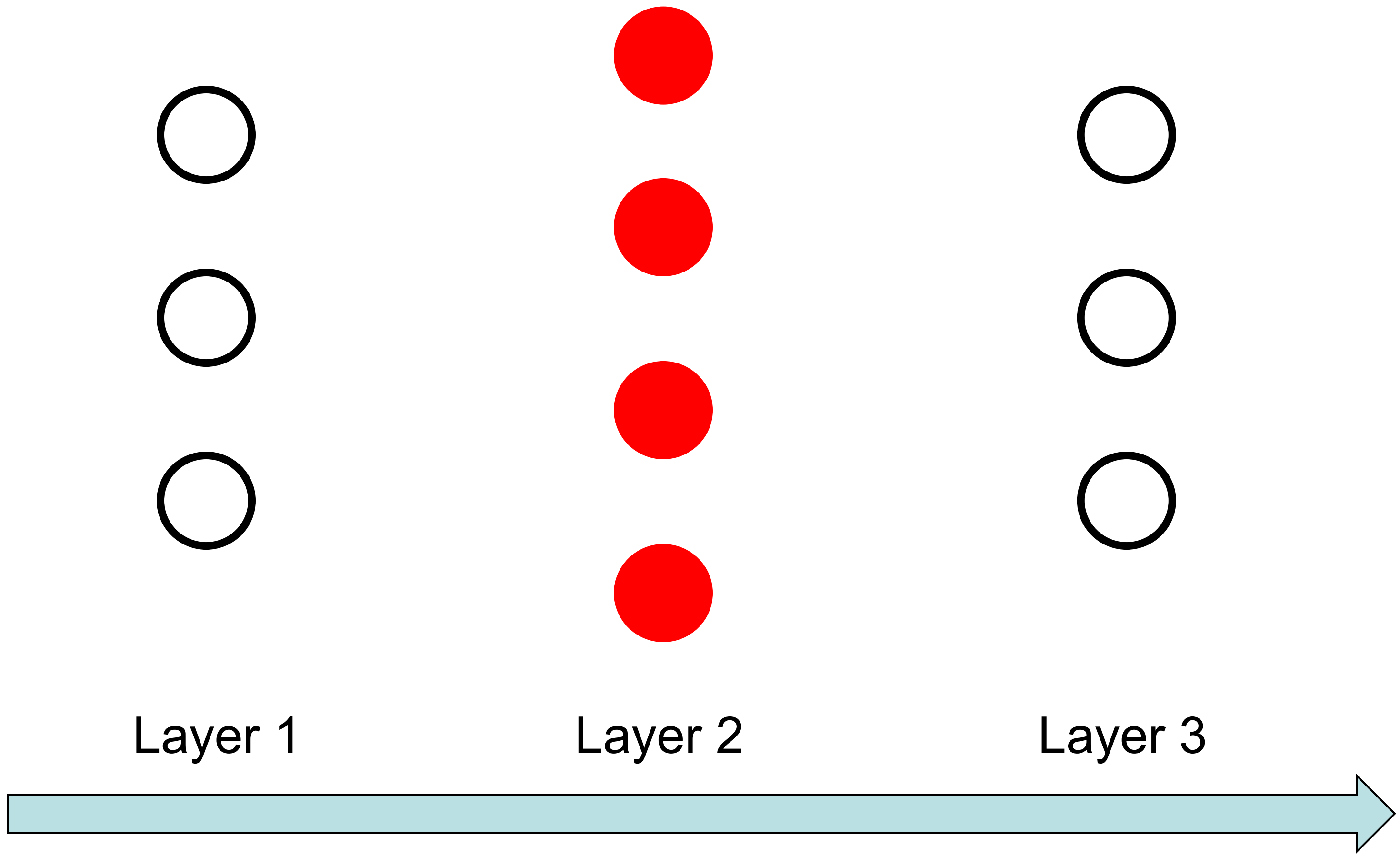
Cagri Ozcinar, PhD



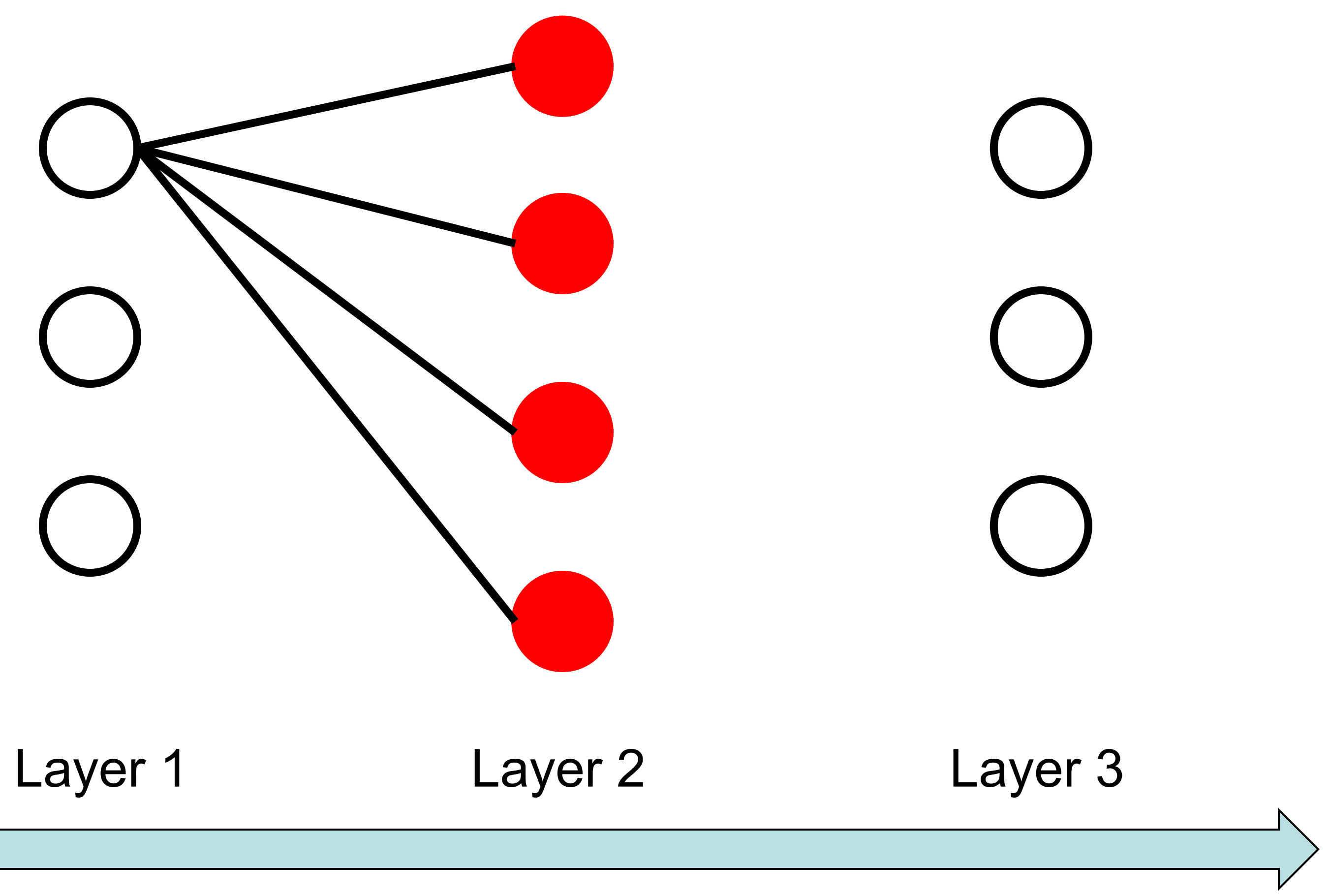
Perceptron model


Input 1
Input 2
Input 3
Input 4
+

Fully-connected Layers

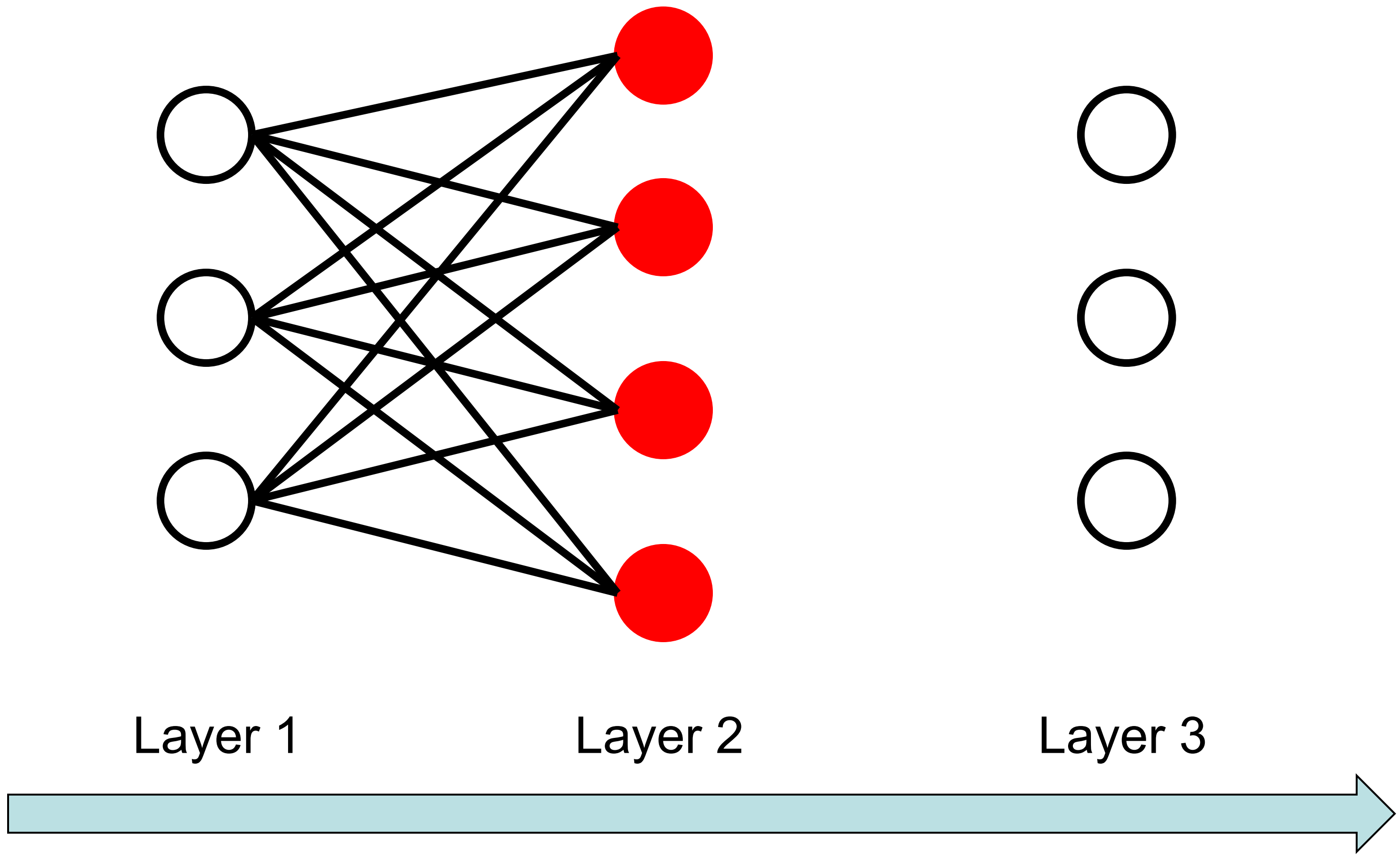


Fully-connected Layers

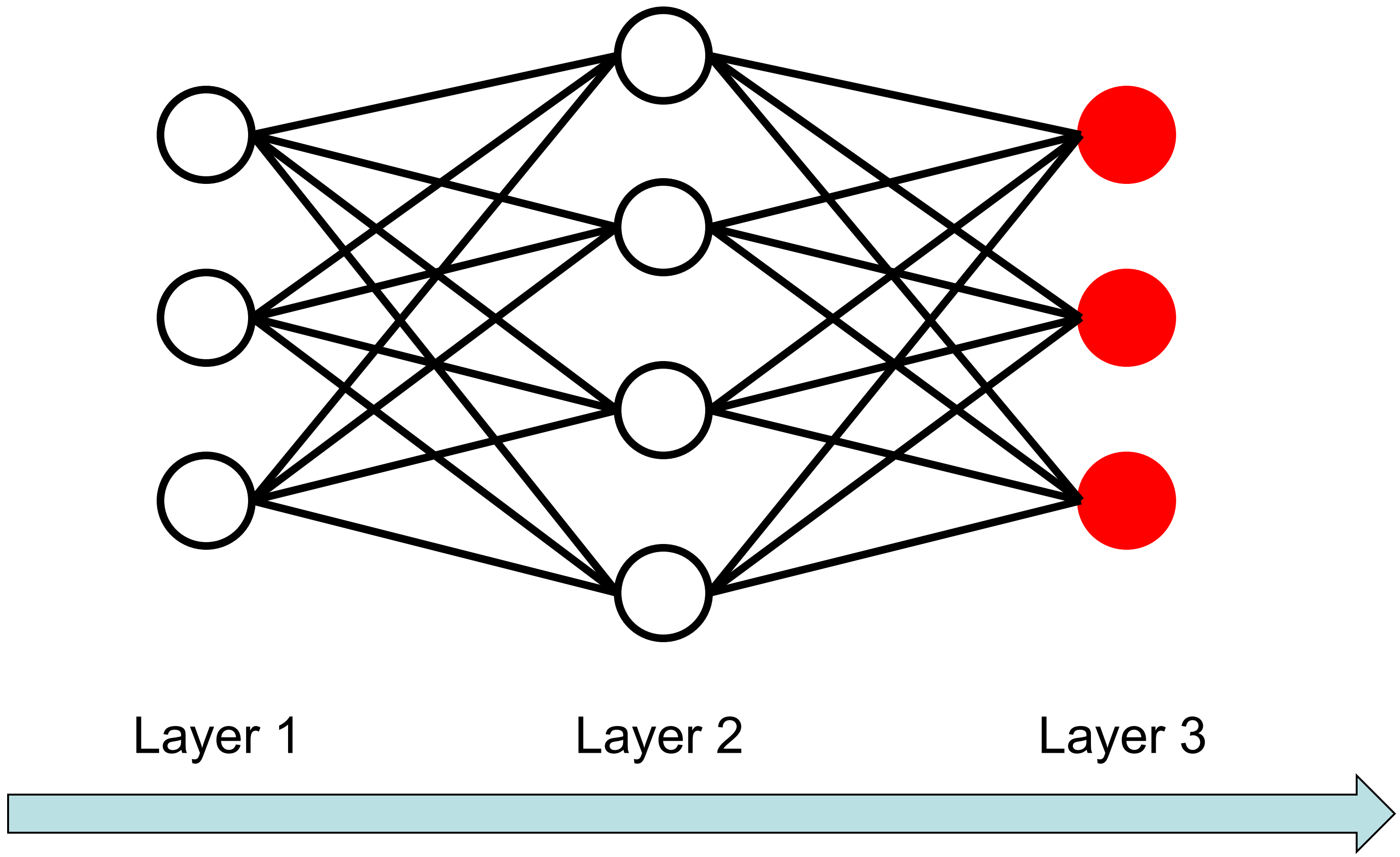


Fully-connected Layers

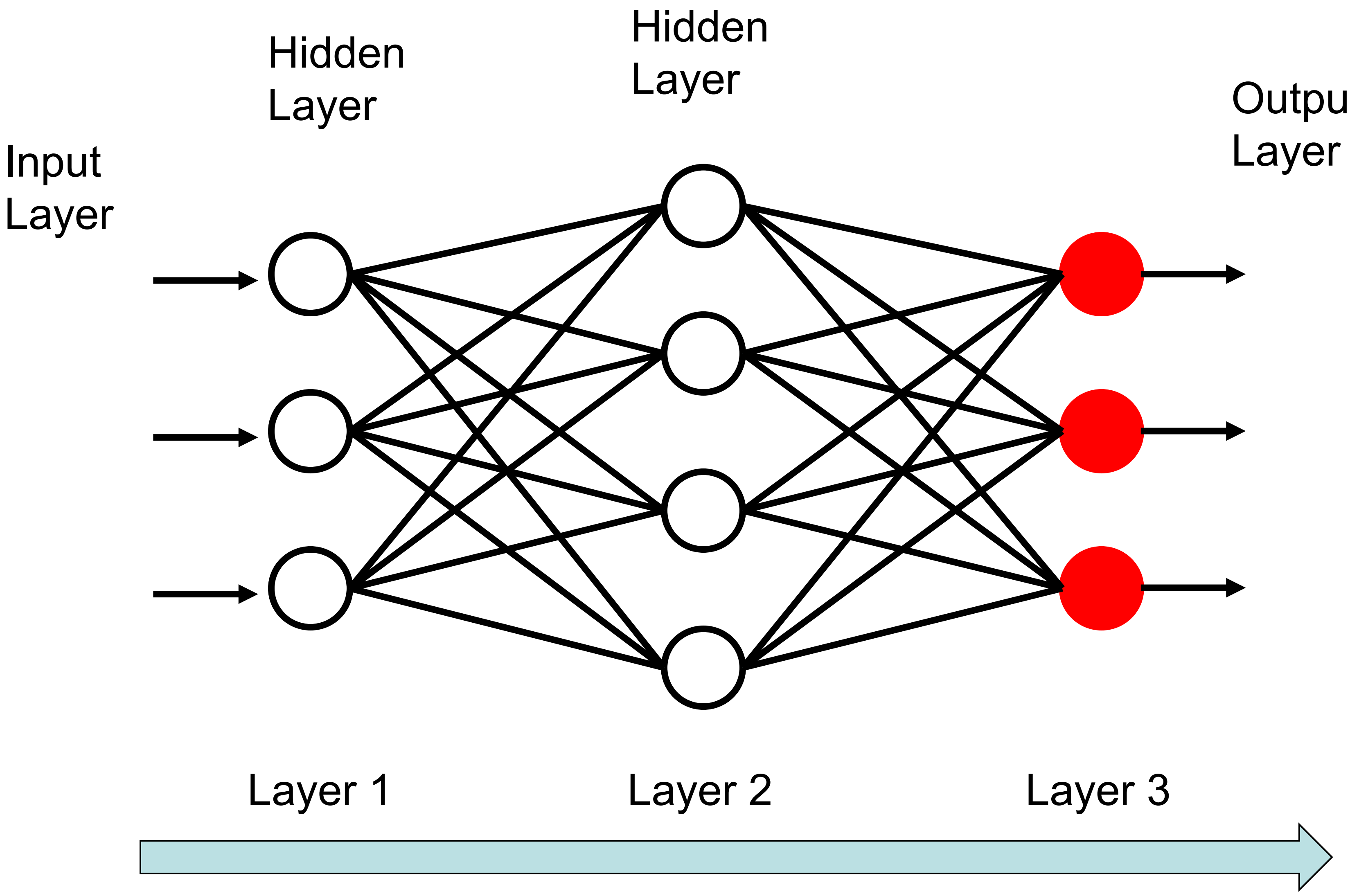
Now we have a fully-connected layer



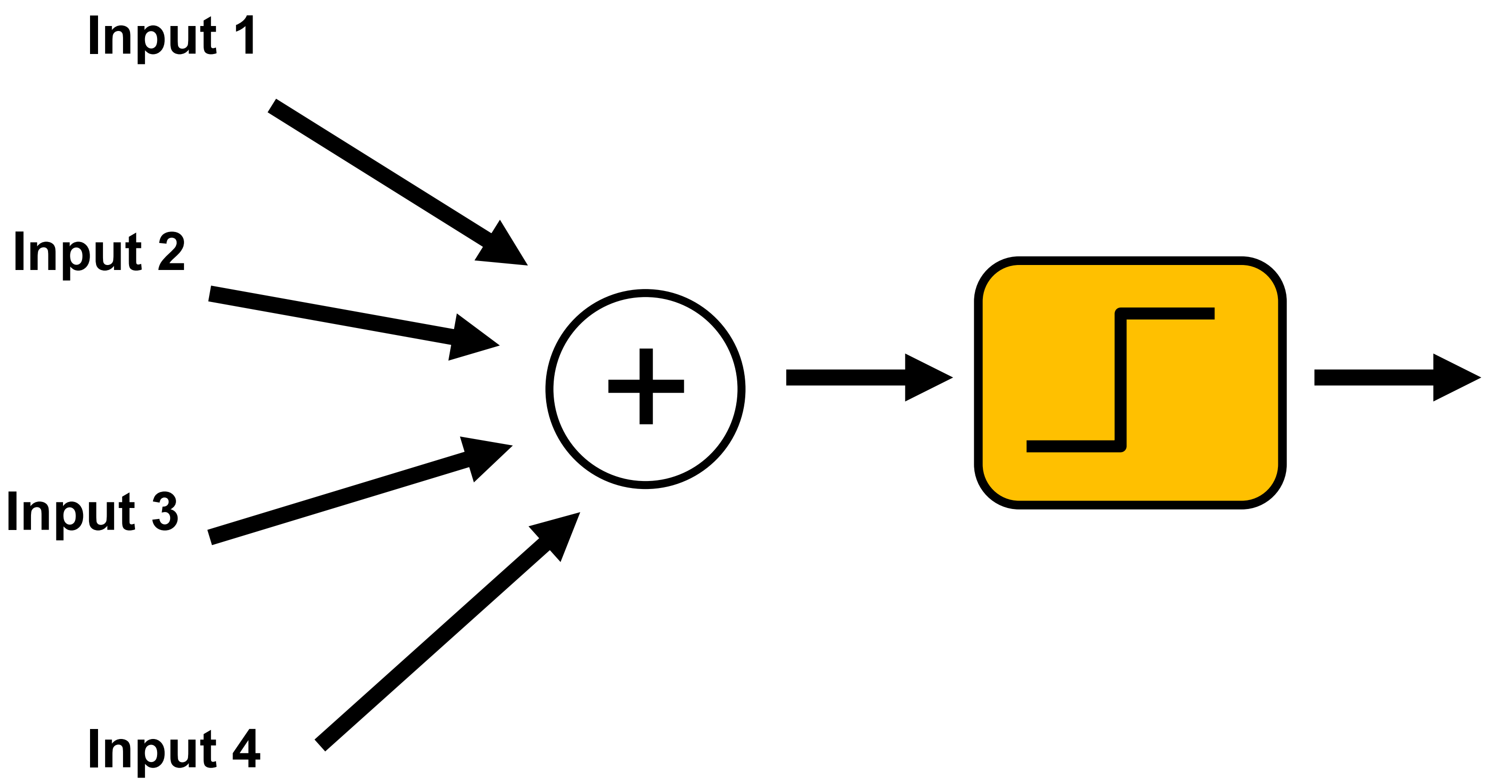
Fully-connected Layers



Fully-connected Layers



Cagri Ozcinar, PhD



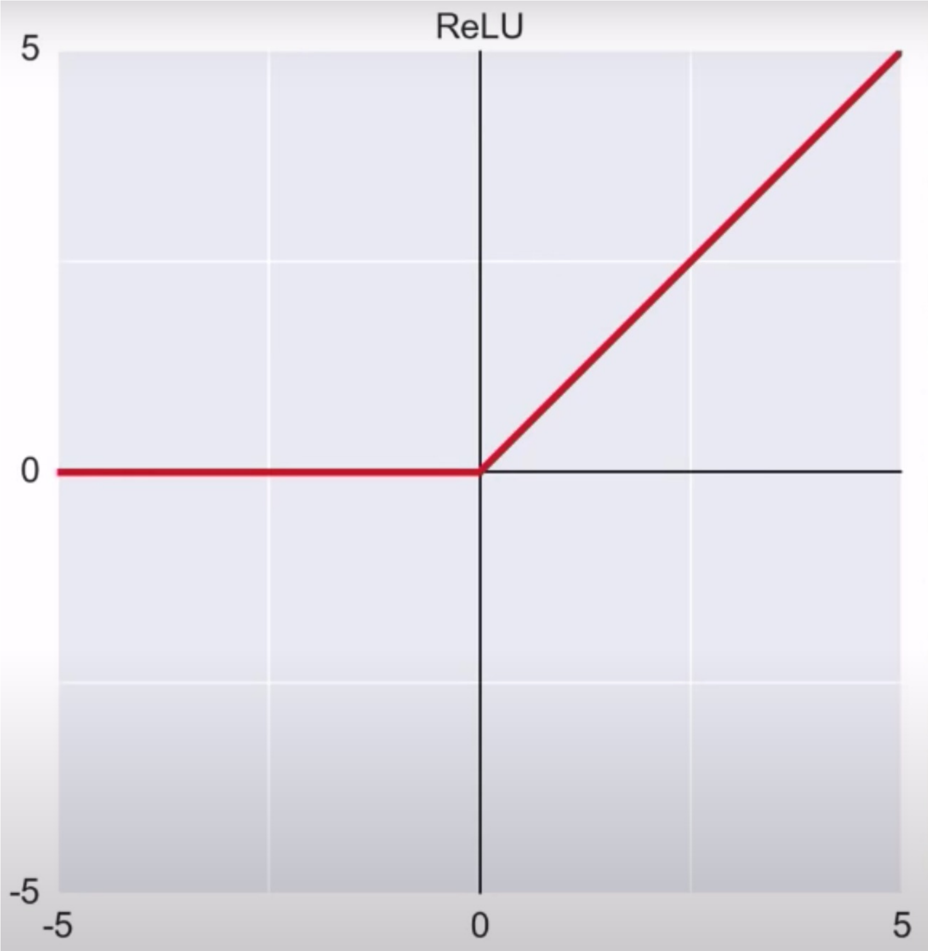
Activation functions

- Rectified Linear Unit (ReLU). The most used activation function.
- Sigmoid. It is especially used for models where we have to predict the probability.
- Hyperbolic tangent activation function (Tanh).

Cagri Ozcinar, PhD

Activation functions

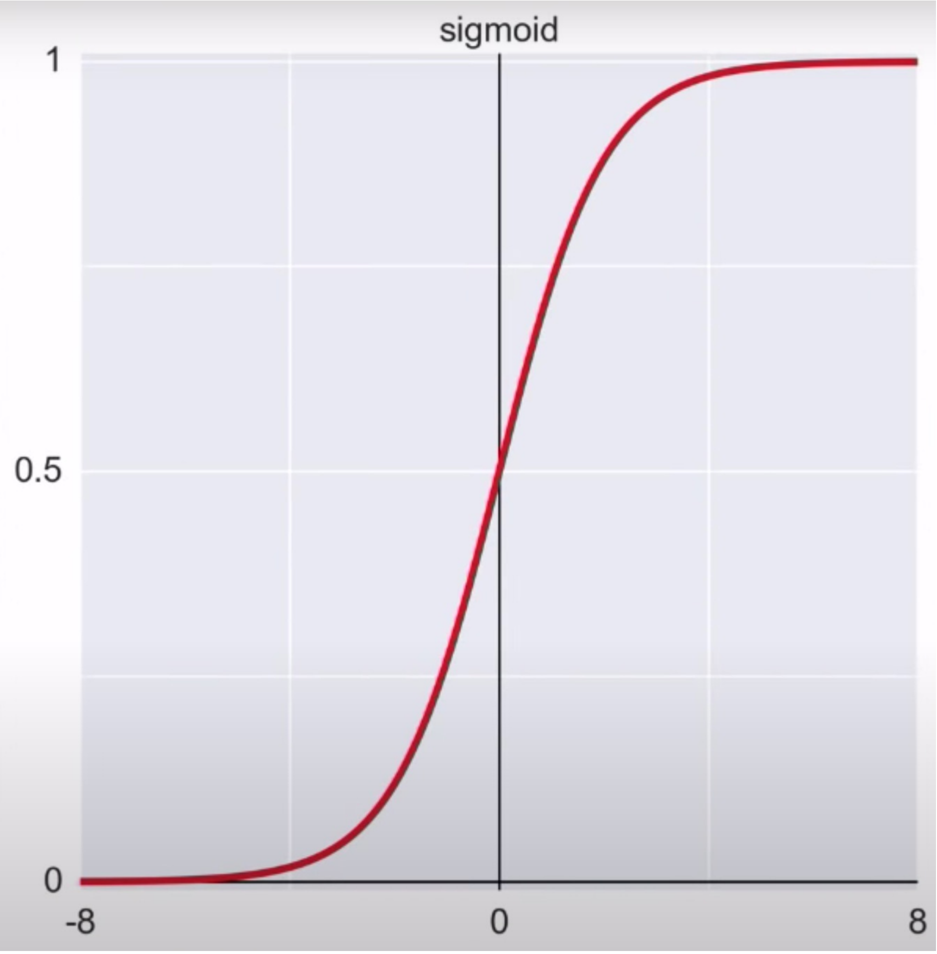
Rectified Linear Unit (ReLU)



Cagri Ozcinar, PhD

Activation functions

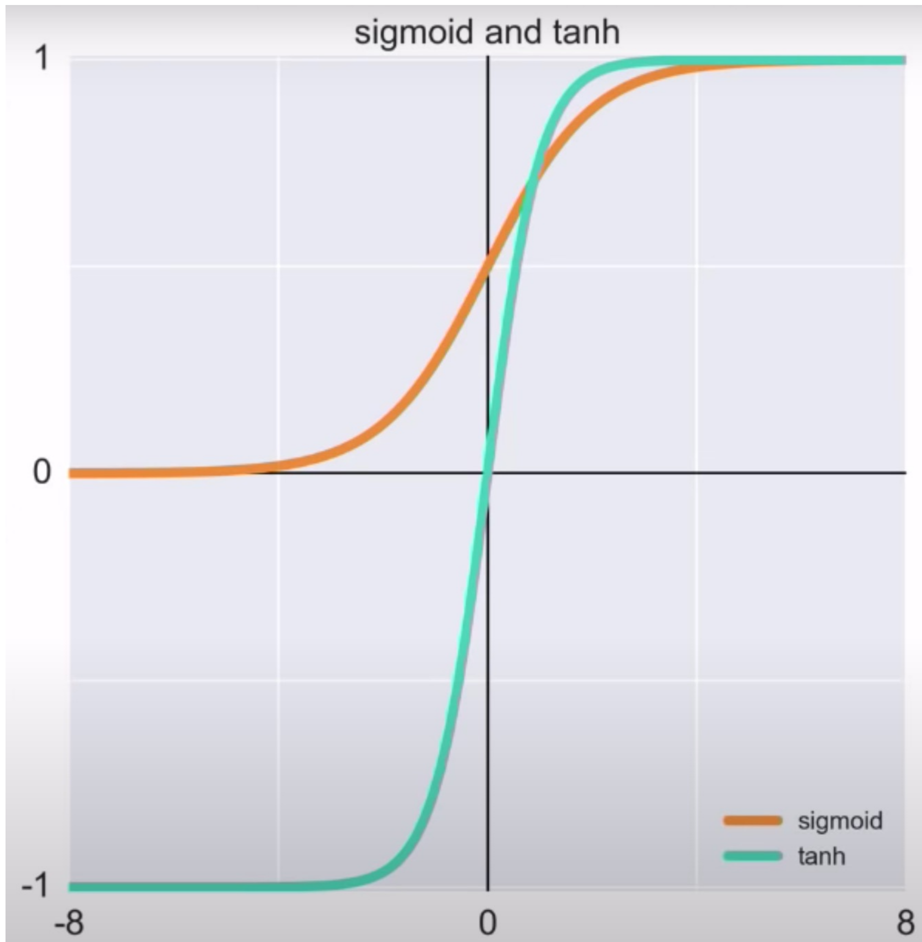
Sigmoid



Cagri Ozcinar, PhD

Activation functions

Hyperbolic tangent activation function (Tanh)



Activation functions

- Any given layer, we assign one activation function to all of the neurons in that layer.

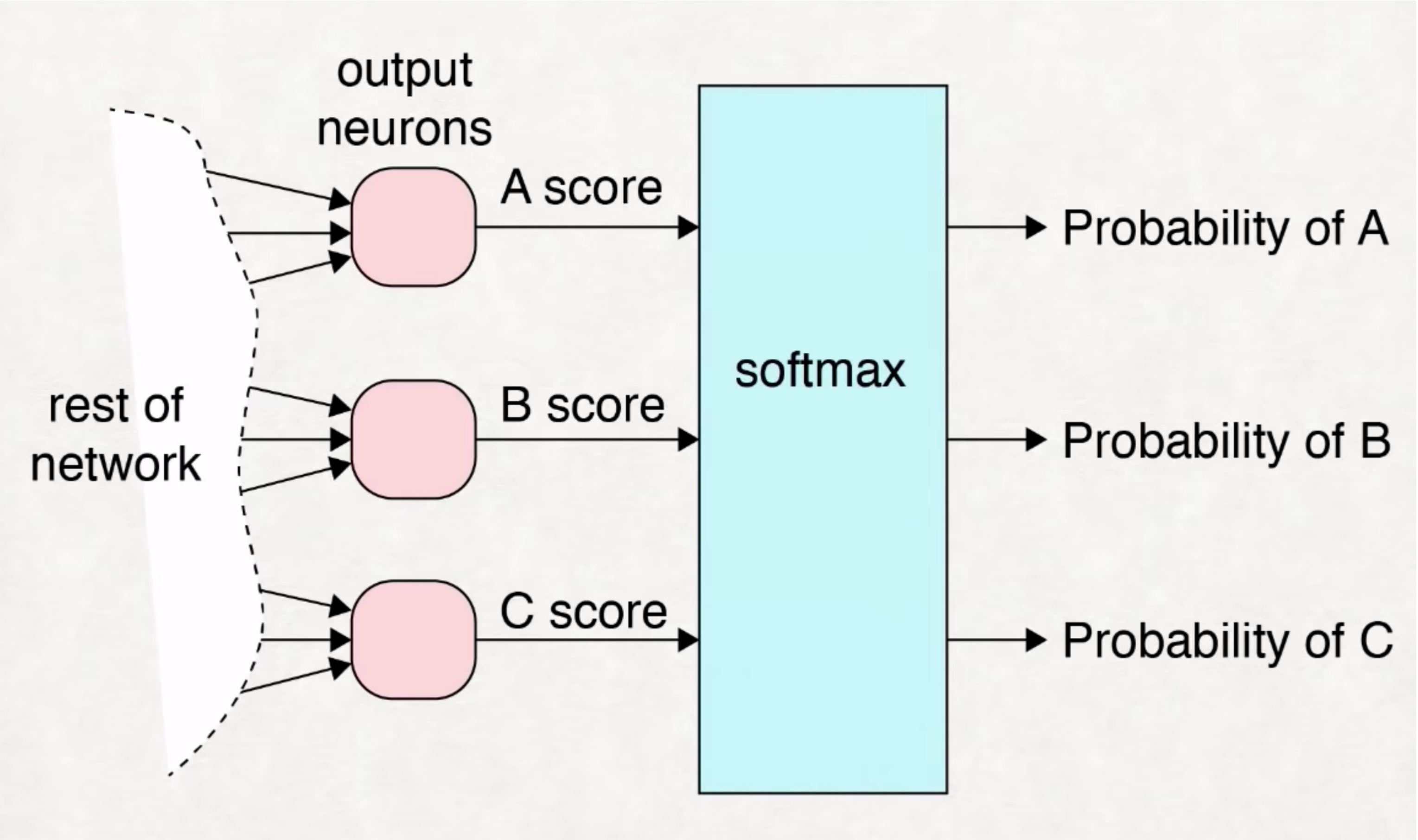
Activation functions

- Softmax

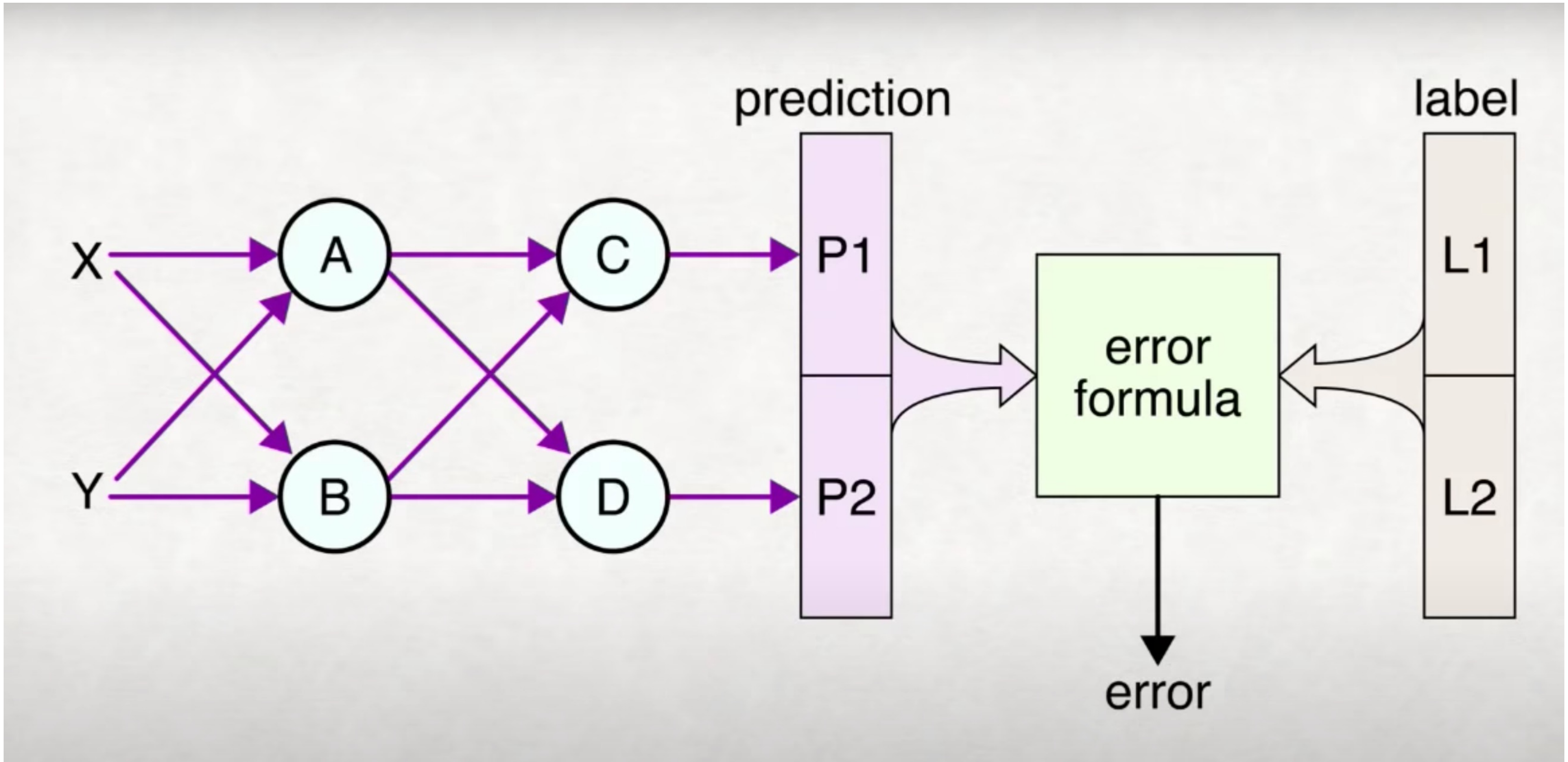
More generalized logistic activation function which is used for multiclass activation

Cagri Ozcinar, PhD

Softmax



Loss functions

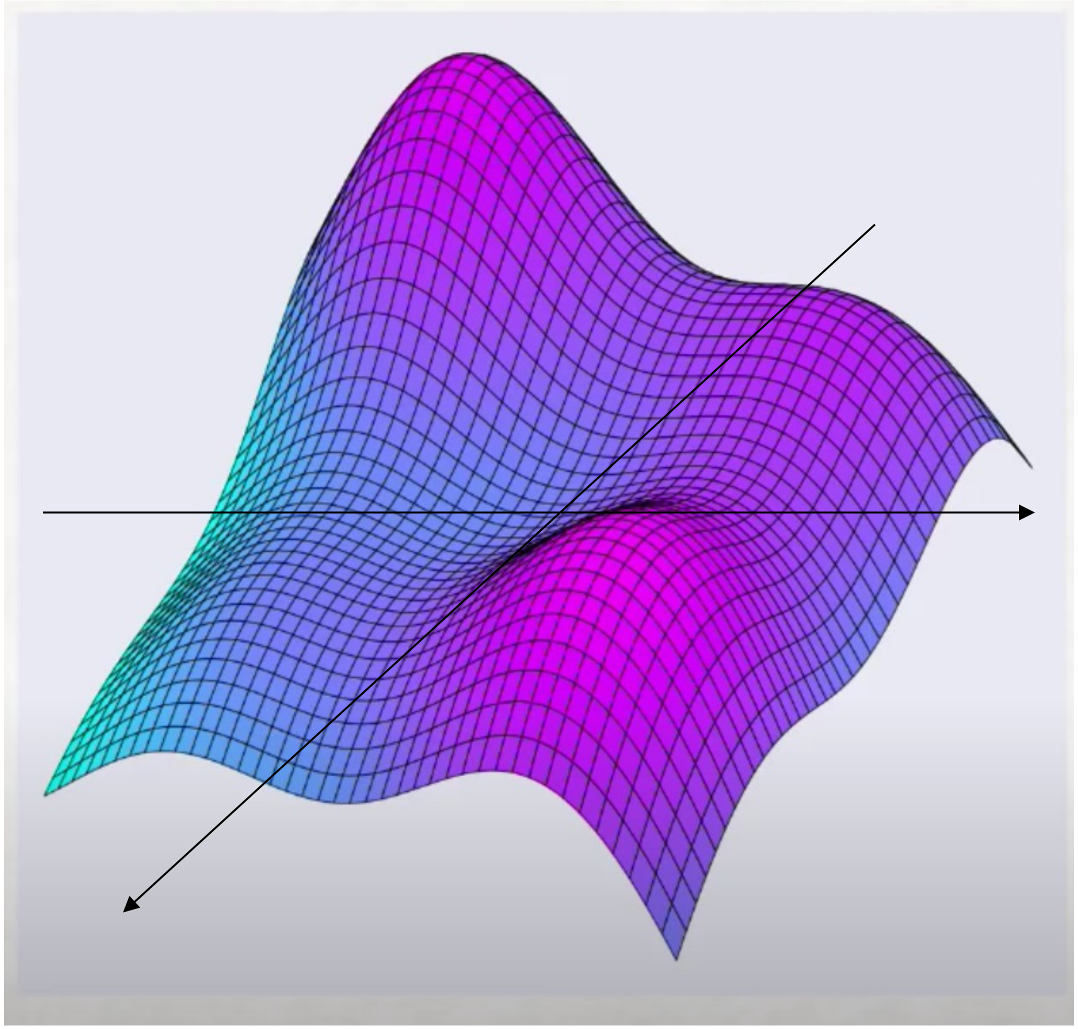


Cagri Ozcinar, PhD

Gradient Decent

- Learning from mistakes

Gradient Decent



Cagri Ozcinar, PhD

Gradient Decent