Introduction to Deep Learning

Tassadaq Hussain

Professor Namal University

Director Centre for AI and Big Data

Collaborations:

Barcelona Supercomputing Center Barcelona, Spain

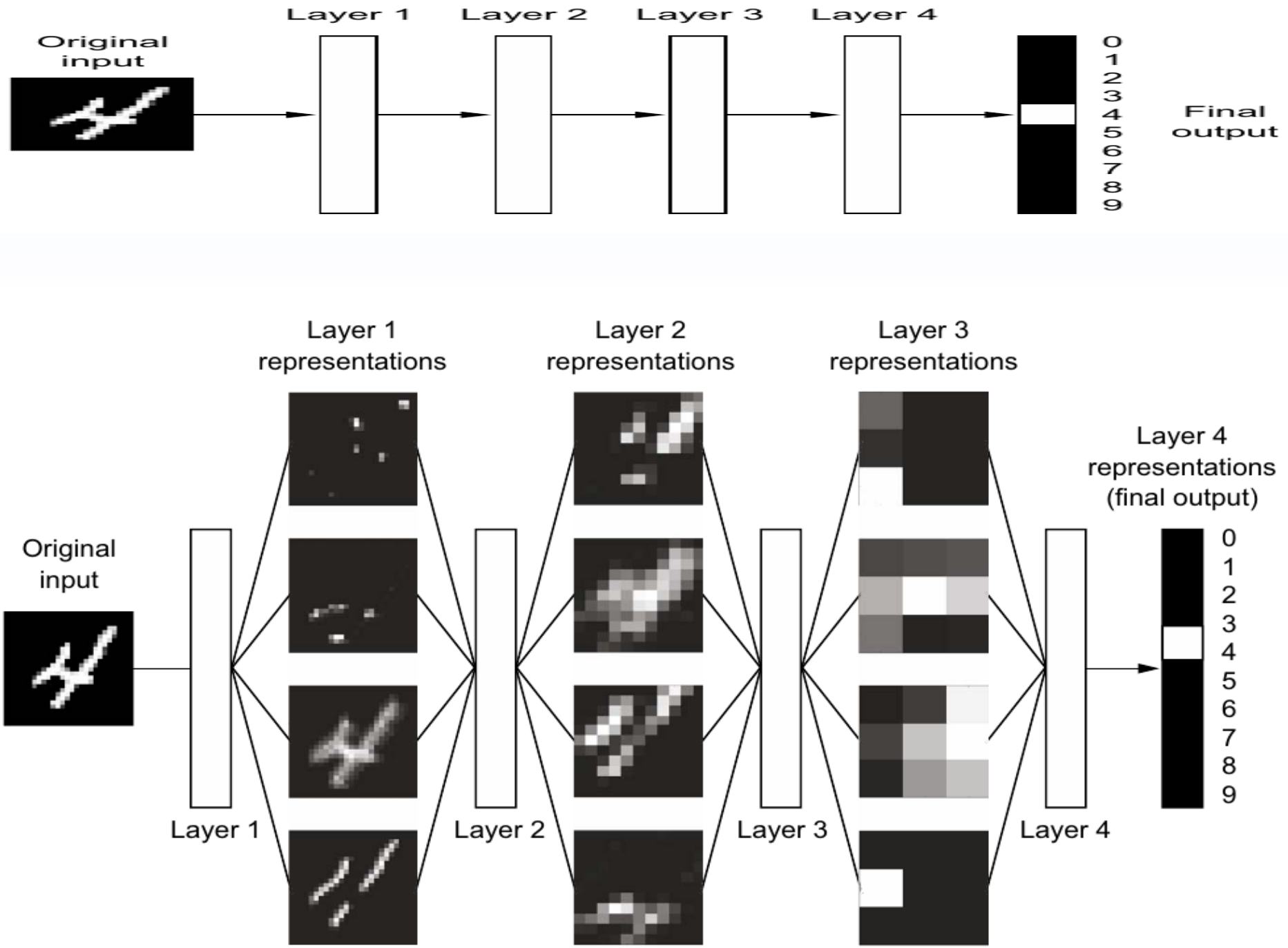
European Network on High Performance and Embedded Architecture and Compilation

Pakistan Supercomputing Center

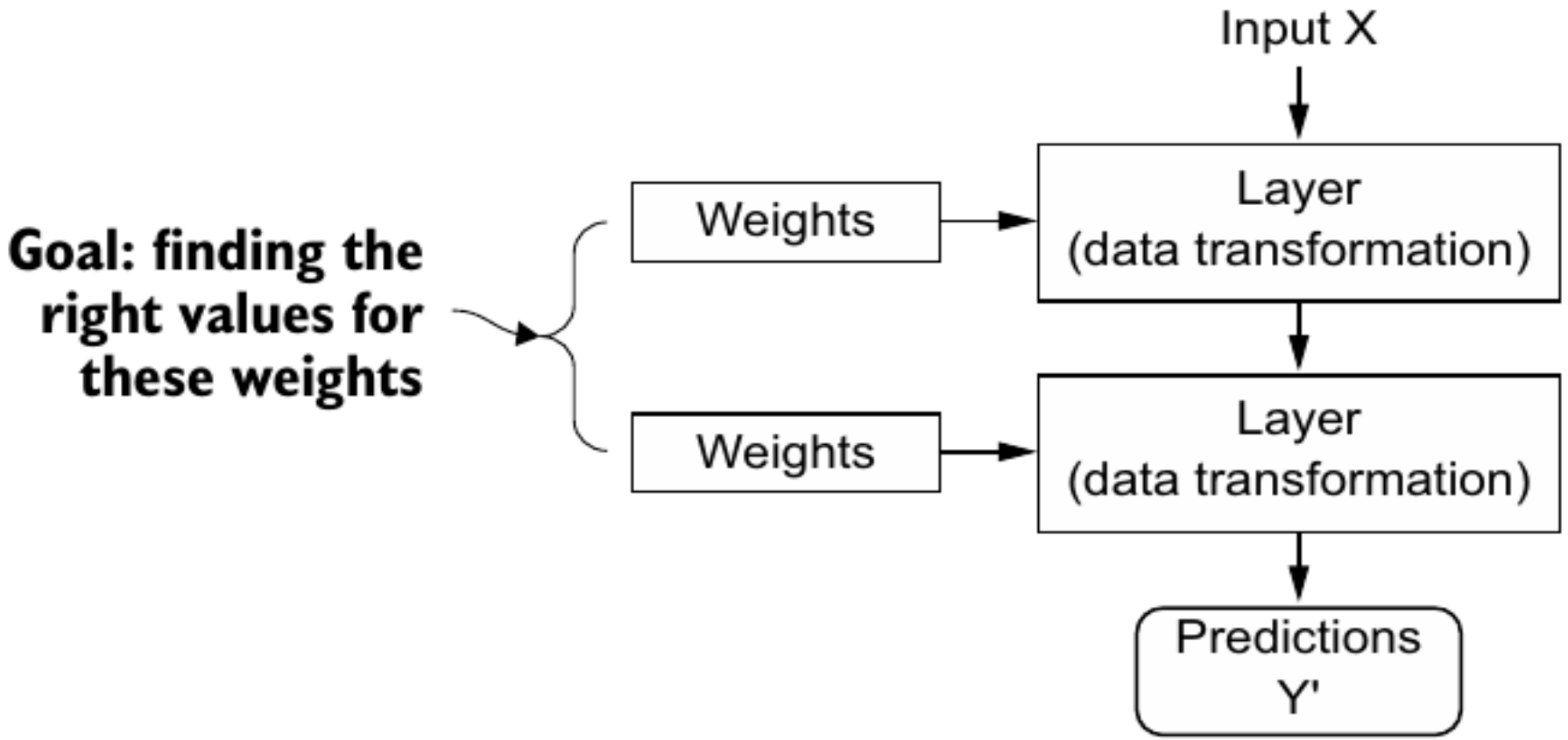
UCERD

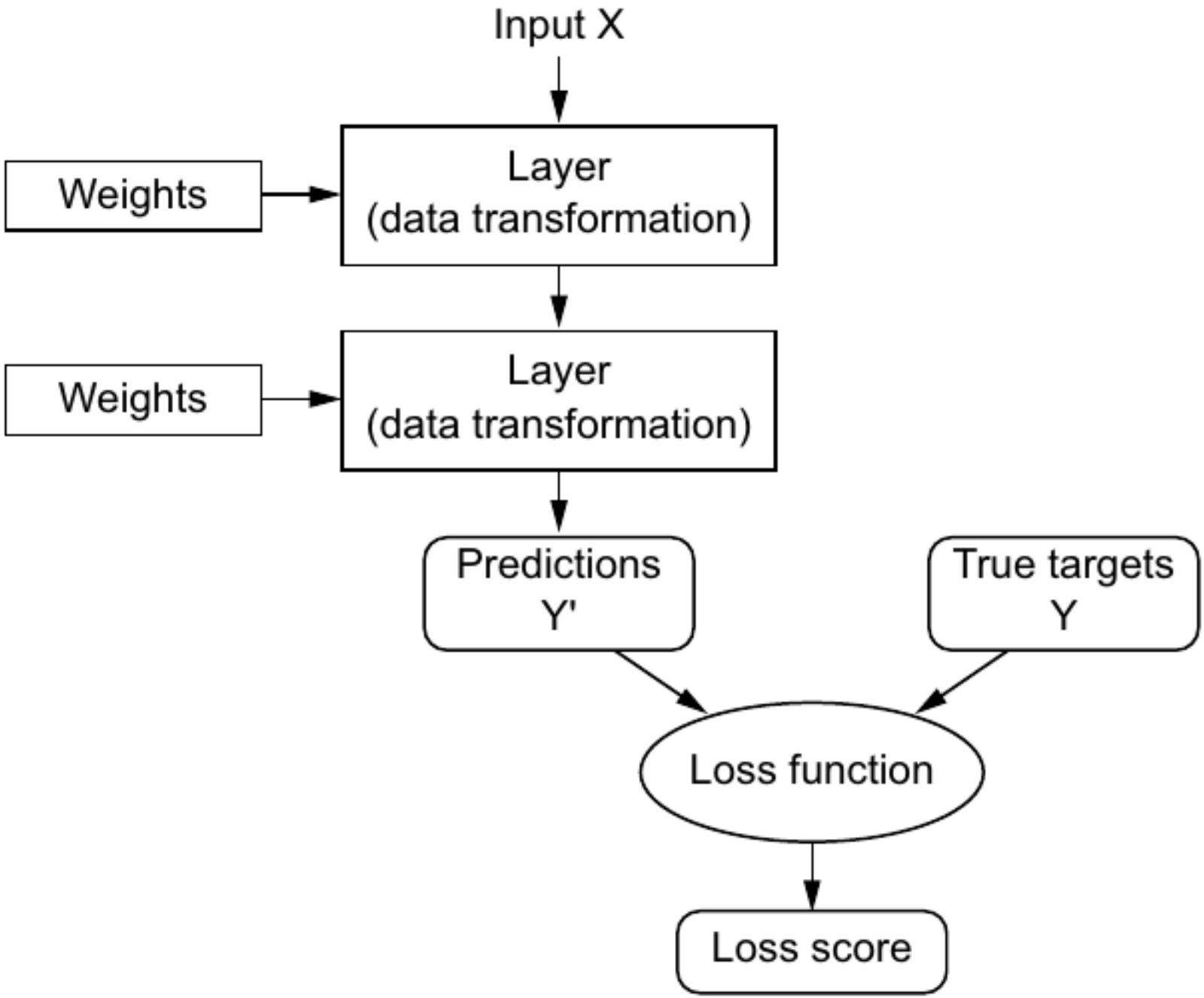
Gathering
Intellectuals

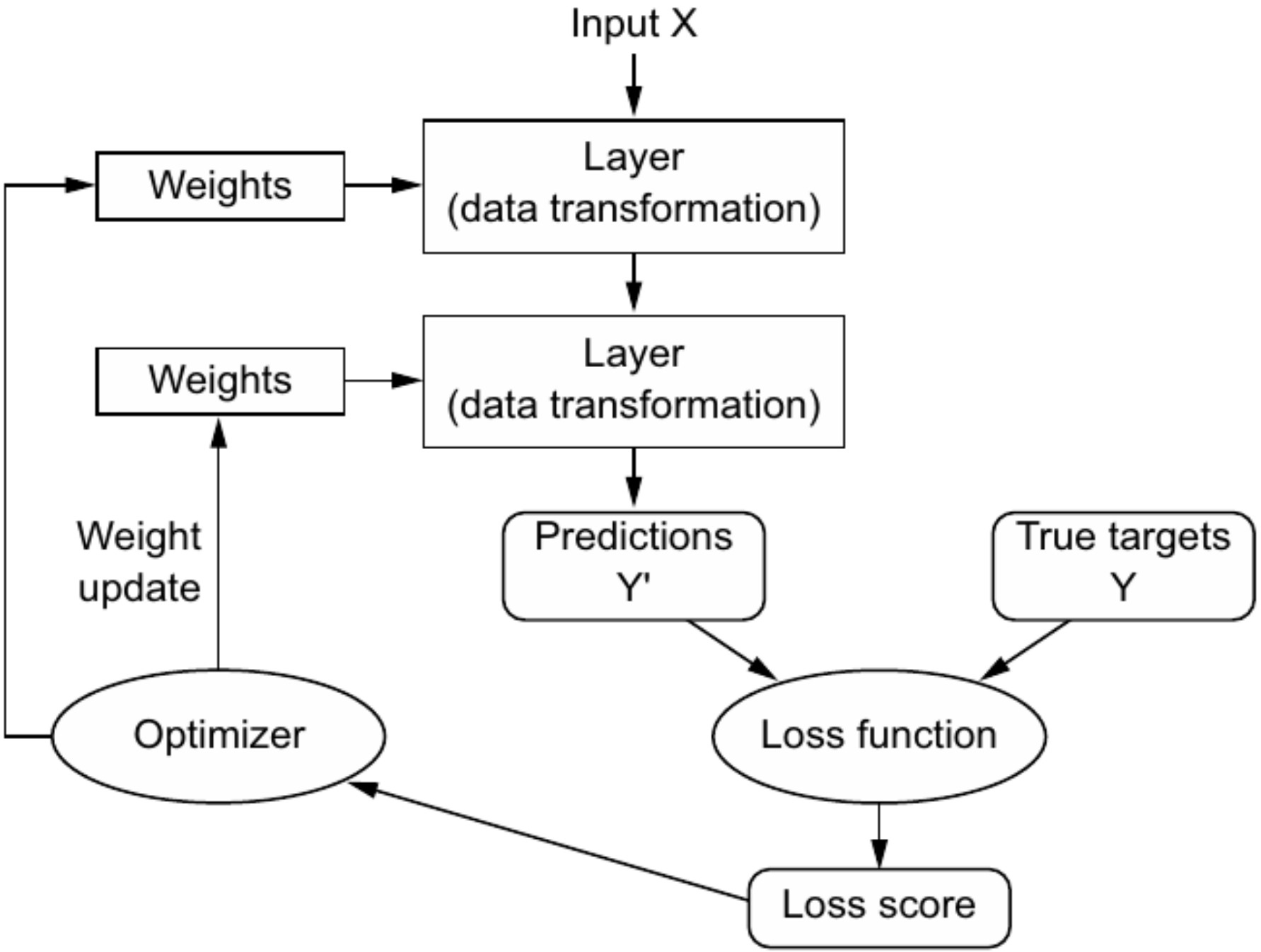
www.ucerd.com


Layer 1
Layer 2
Layer 3
Layer 4
Original input
Final output
0
1
2
3
4
5
6
7
8
9
Layer 1 representations
Layer 2 representations
Layer 3 representations
Layer 4 representations (final output)
Original input
Layer 1
Layer 2
Layer 3
Layer 4
0
1
2
3
4
5
6
7
8
9

Understanding how deep learning works




Input X
Weights
Layer
(data transformation)
Weights
Layer
(data transformation)
Predictions
Y'
True targets
Y
Loss function
Loss score

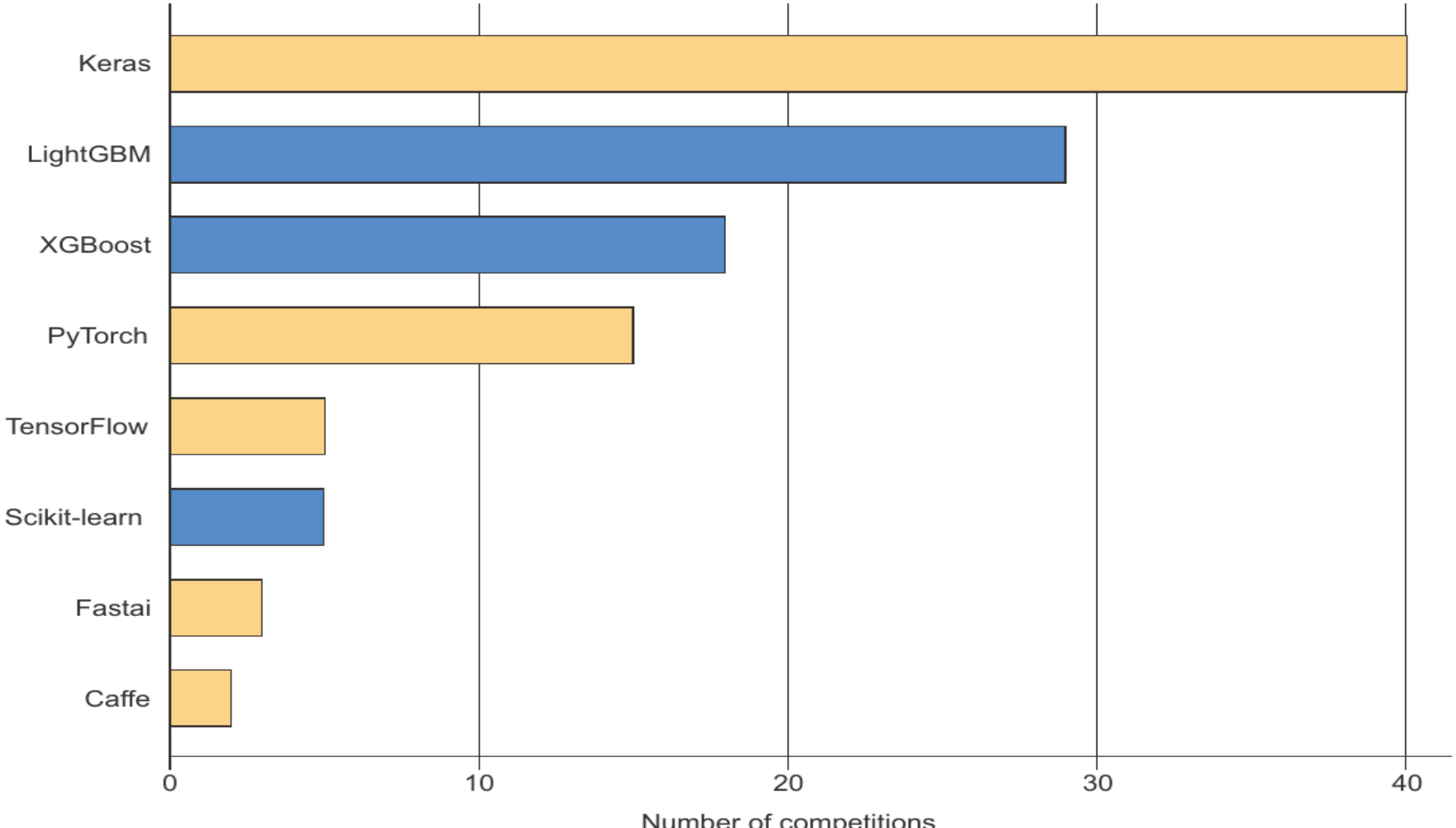

Input X
Weights
Layer
(data transformation)
Weights
Layer
(data transformation)
Weight
update
Predictions
Y'
True targets
Y
Optimizer
Loss function
Loss score

What deep learning has achieved so far

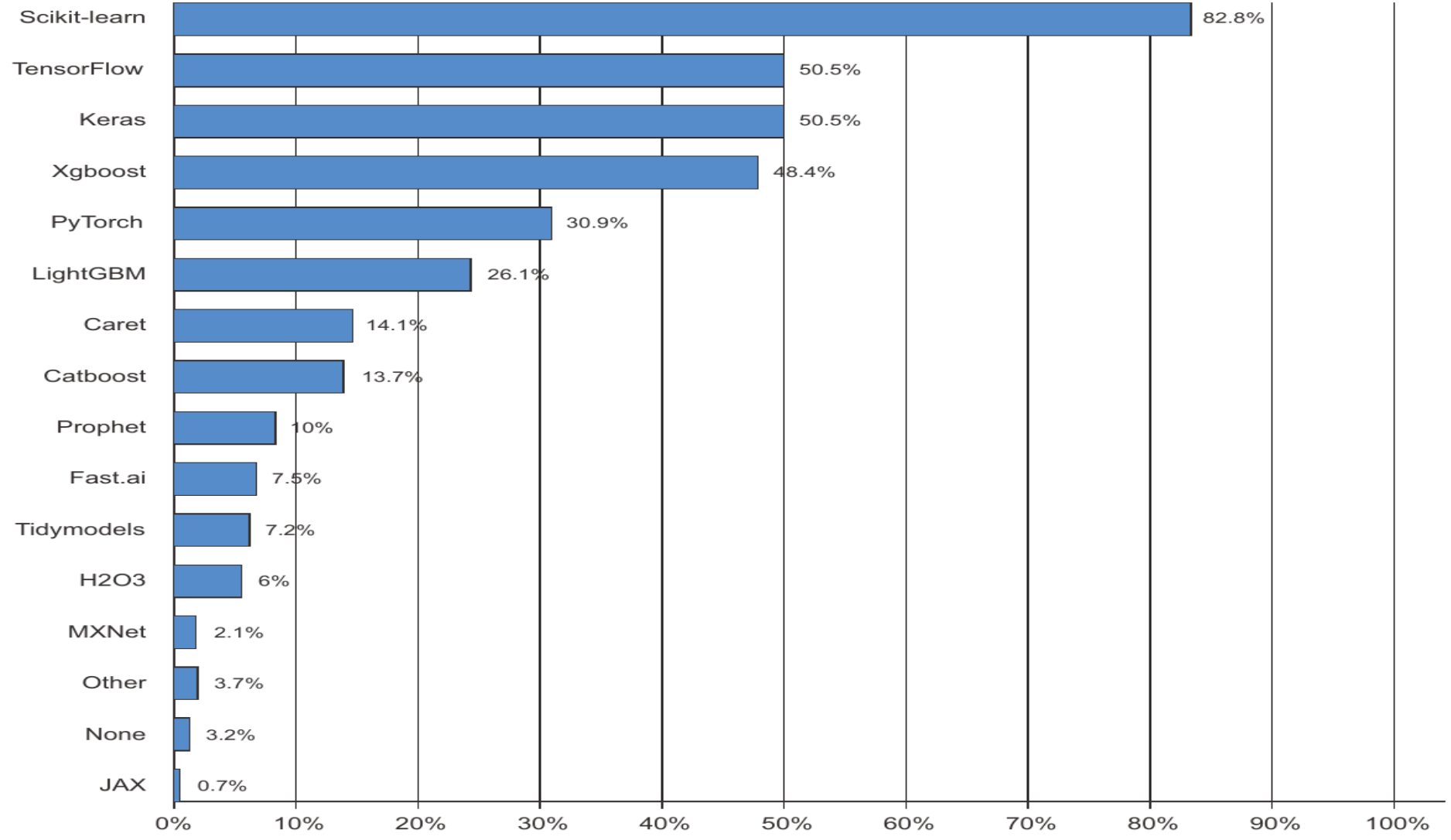
- Near-human-level image classification
- Near-human-level speech transcription
- Near-human-level handwriting transcription
- Dramatically improved machine translation
- Dramatically improved text-to-speech conversion
- Digital assistants such as Google Assistant and Amazon Alexa
- Near-human-level autonomous driving
- Improved ad targeting, as used by Google, Baidu, or Bing
- Improved search results on the web
- Ability to answer natural language questions
- Superhuman Go playing

UCERD

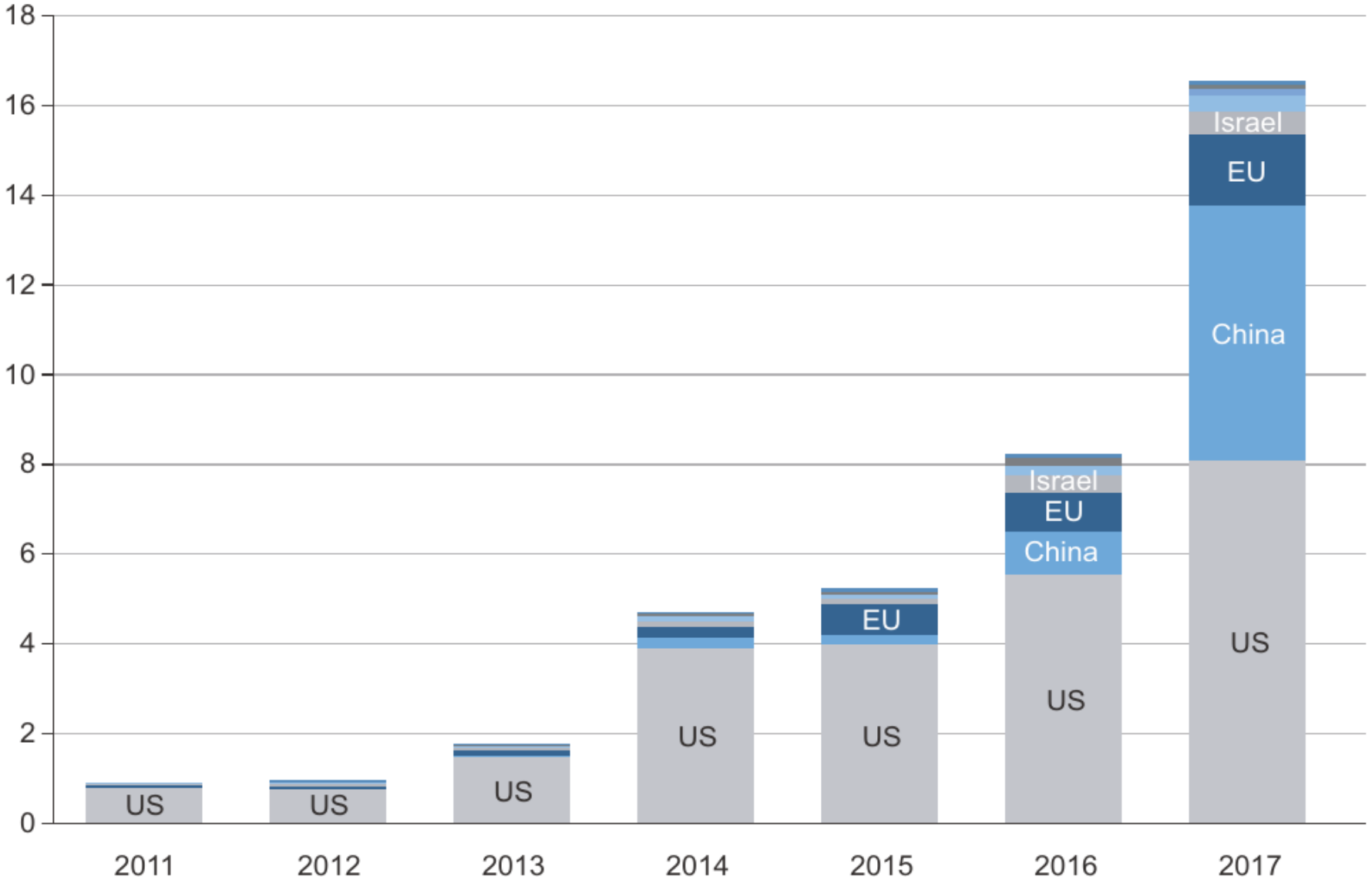
The modern machine learning landscape



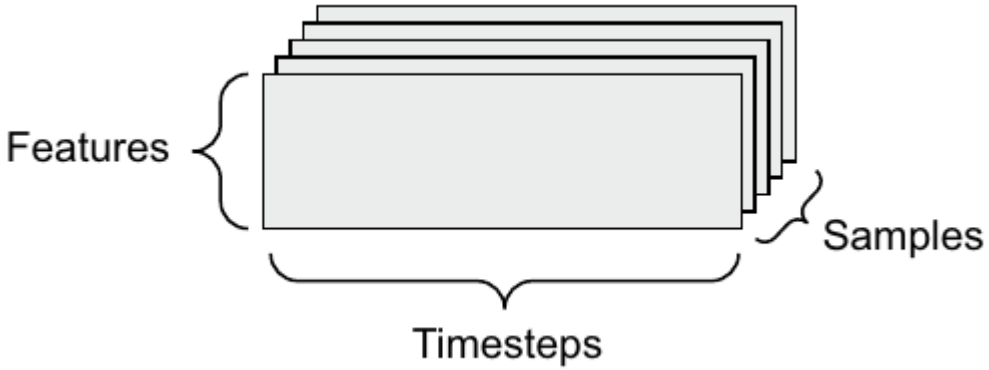
Tool usage across the machine learning and data science industry

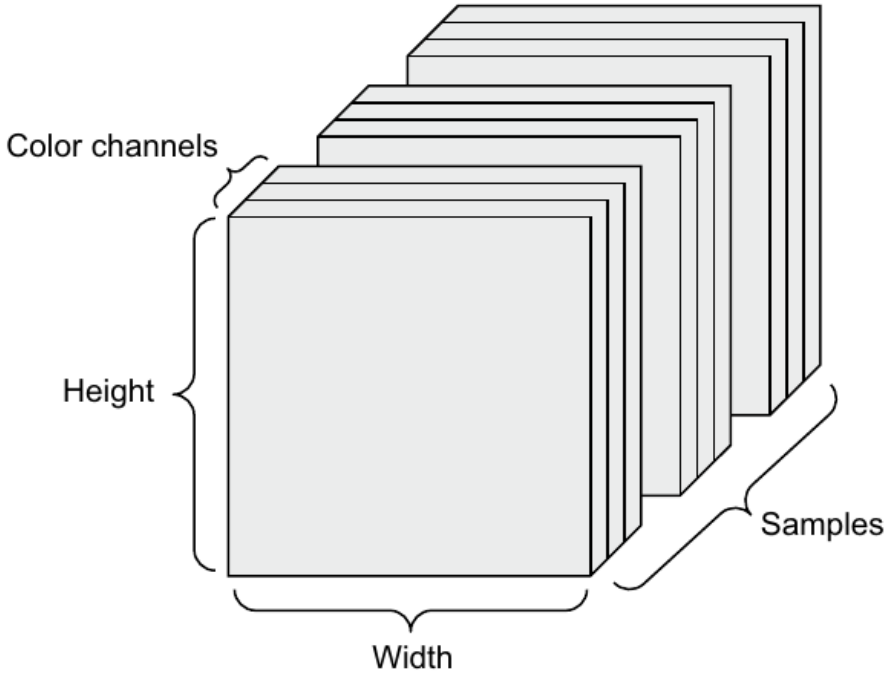


Investments in AI based Startups

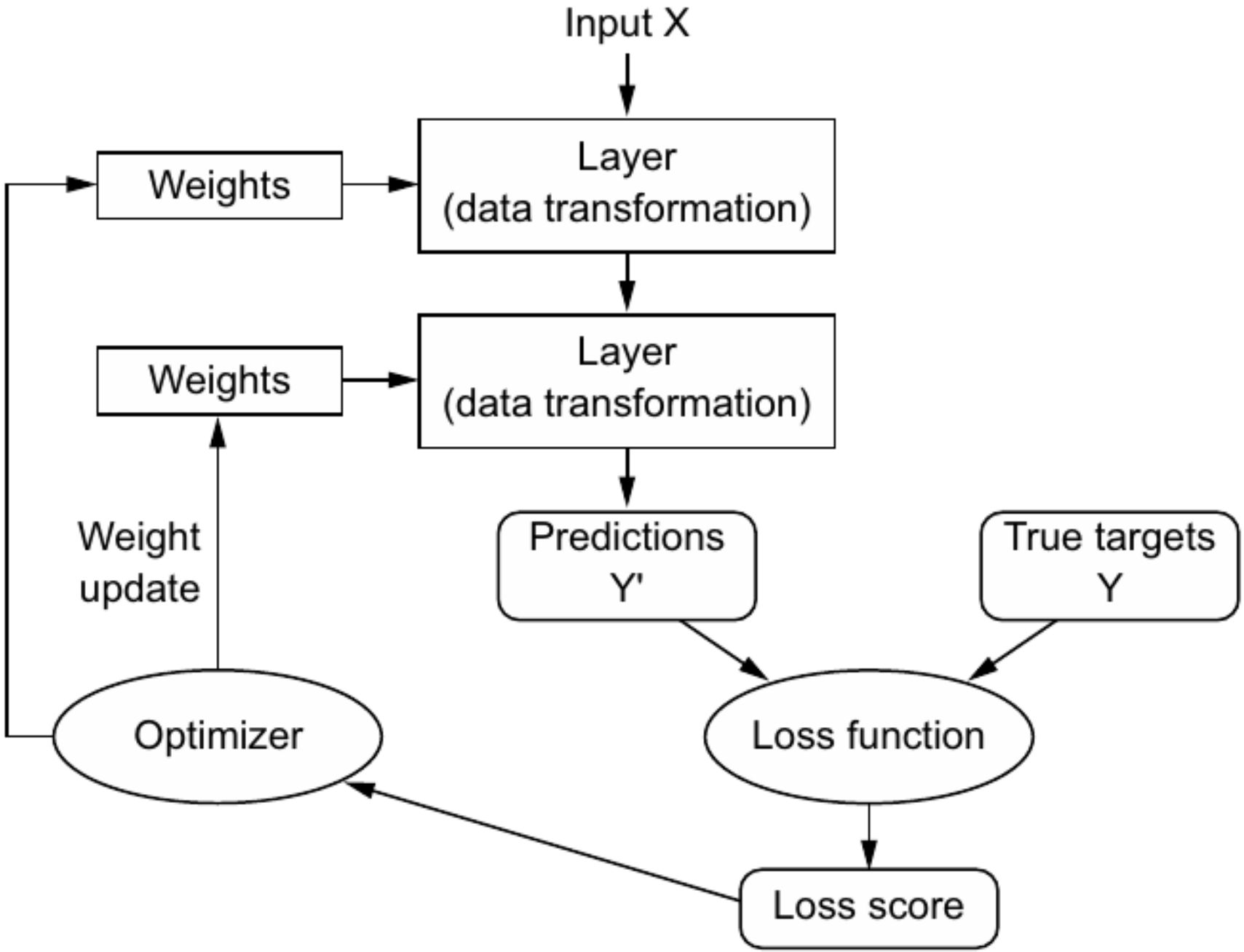


Understand Data





UCERD


Input X
Weights
Layer
(data transformation)
Weights
Layer
(data transformation)
Weight
update
Predictions
Y'
True targets
Y
Optimizer
Loss function
Loss score

from tensorflow.keras.datasets import mnist
(train_images, train_labels), (test_images, test_labels) = mnist.load_data()

```
(train_images, train_labels), (test_images, test_labels) = mnist.load_data()
train_images = train_images.reshape((60000, 28 * 28))
train_images = train_images.astype("float32") / 255
test_images = test_images.reshape((10000, 28 * 28))
test_images = test_images.astype("float32") / 255
```

```
model = keras.Sequential([
    layers.Dense(512, activation="relu"),
    layers.Dense(10, activation="softmax")
])
```

```
model.compile(optimizer="rmsprop",
              loss="sparse_categorical_crossentropy",
              metrics=["accuracy"])
```

```
model.fit(train_images, train_labels, epochs=5, batch_size=128)
```