Deep Learning

Deep Learning: What is it?

Deep learning has seen some of the most spectacular developments in artificial intelligence recently. Deep learning models have come close to or even surpassed human performance on tasks like natural language translation, image recognition, and gaming.

What then is deep learning? Deep stacks of computations characterize the deep learning method of machine learning. Deep learning models have been able to decipher the intricate and hierarchical patterns present in the most difficult real-world datasets because to this depth of processing.

The defining deep learning model is now neural networks due of their strength and scalability. Neurons make up neural networks, and each neuron in a neural network can only do a single simple calculation.

The Linear Unit

So, let's begin with the fundamental component of a neural network: the individual neuron. As a diagram, a neuron (or unit)

from tensorflow import keras

from tensorflow.keras import layers

# Create a network with 1 linear unit

model = keras.Sequential([

    layers.Dense(units=1, input\_shape=[3])

])

model.summary()

………………………………………………………………………………………………………………………………………………………………………………………….

neural networks typically organize their neurons into **layers**

An **activation function** is simply some function we apply to each of a layer's outputs (its *activations*). The most common is the *rectifier* function max(0,x)max(0,x).

The layers before the output layer are sometimes called **hidden** since we never see their outputs directly

from tensorflow import keras

from tensorflow.keras import layers

model = keras.Sequential([

*# the hidden ReLU layers*

layers.Dense(units=4, activation='relu', input\_shape=[2]),

layers.Dense(units=3, activation='relu'),

*# the linear output layer*

layers.Dense(units=1),

])

model.summary()

………………………………………………………………………………………………………………………………………………………………………………………………

The **loss function** measures the disparity between the the target's true value and the value the model predicts.

The optimizer is an algorithm that adjusts the weights to minimize the loss.

model.compile(

optimizer="adam",

loss="mae",

)

…………………………………………………………………………………………………………………………………………………………………………………………………