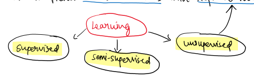
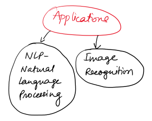
FLOCK-DAY 2

- What is Deep Learning?

↳ It is a part of a broader family of machine learning methods based on artificial neural networks with representation learning.



→ set of techniques that allows a system to automatically discover the representations needed for feature detection of raw data.



- What are ANNs or NNs?

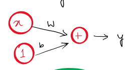
↳ They are computing system vaguely inspired by biological neural networks that constitute animal brain.

↳ ANN is based on a collection of units or nodes called artificial neurons.

→ Each neuron individually performs only a single computation.

THE LINEAR UNIT

↳ Individual Component of Neural Network: A neuron.
It would look something like this:



The linear unit: $y = wx + b$ → Does this look familiar?

where—

- x is the input.
- w is the weight (It is associated to the weight of connections)
- b is a special kind of weight called the bias (it does not have any input data)
- y is the ultimate output of the neuron (it sums up all the values received)

* SINGLE NEURON MODELS ARE LINEAR MODELS

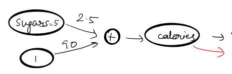
For example:
Let's have a dataset of 80 cereals.
Input → Grams of sugars per serving - "sugars"
Output → Calories per serving - "calories"

$b = 90$
$w = 2.5$ } Let's assume

Bias → Difference b/w the expected value of the estimator and parameter.

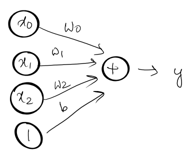
Weight → trainable and untrainable parameters of a model.

Using the given info, we can estimate the calorie content of the cereal.



$b + wx$
$\rightarrow 90 + (5 \times 2.5)$
$= 102.5$

MULTIPLE INPUTS



$y = w_0x_0 + w_1x_1 + w_2x_2 + b$

- What is keras?

→ Open source software library that provides a Python interface for artificial neural networks

LINEAR UNITS IN KERAS

- How to create a Model in Keras?

→ keras.Sequential → creates a neural network as a stack of layers.

Let's look at the code:

```
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])
])
```

* units → no. of outputs = 1 (in our case)

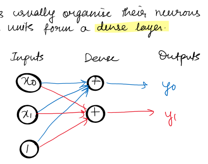
* input_shape → dimensions of the inputs

DEEP NEURAL NETWORKS

In this lesson, we will learn how we can build neural networks capable of learning the complex kinds of relationships deep neural networks are famous for.

Layers

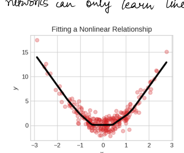
- Neural networks usually organise their neurons in layers.
- Multiple linear units form a dense layer.



- Through a deep stack of layers, a neural network can transform its inputs in more and more complex ways.

THE ACTIVATION ENERGY

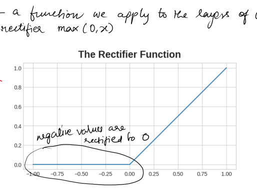
- As it seems, two dense layers with nothing in between is no better than a single dense layer.
- We need something non-linear to make sure dense layers can move into the world of lines of planes.
- That is where Activation Energy comes in.
- Without AE, neural networks can only learn linear relationships.



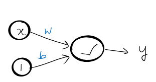
ACTIVATION FUNCTION – a function we apply to the layers of outputs.

- Most common – rectifier max $(0, x)$

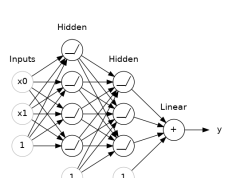
→ provides a link/interaction between several dense layers.



- When we attach the rectifier to a linear unit, we get a rectified linear unit or ReLU. (For this reason, it's common to call the rectifier function the "ReLU function".) Applying a ReLU activation to a linear unit means the output becomes max(0, w * x + b), which we might draw in a diagram like:



STACKING DENSE LAYERS



BUILDING SEQUENTIAL MODELS

```
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),
])
```

STOCHASTIC GRADIENT DESCENT

So far, we saw what neurons and neural networks are.
We worked with pre-defined values of bias and weights.
The network does not know anything yet.
Now, we are going to see how can we train the model.

In addition to training the network with data, we need—

1) A loss function
2) An optimizer

THE LOSS FUNCTION

- The loss function measures the disparity between the targets true value and the value the model predicts.
- Different problems have different loss functions.
- For regression problems, the mean absolute error or MAE is the common loss function.

THE OPTIMIZER

What we have done so far—
* Described the problem we want the network to solve.
Now, we want to say how we want it to be solved.
That's the job of the optimizer.

In a way, we can say all deep learning algorithms belong to the "stochastic gradient descent."

- Each iteration's sample of training data is called a minibatch.
- A complete round of training data is called epoch.

LETS LOOK AT AN EXAMPLE!