Lesson 2

Week 1

Lesson 1 : Fundamentals

Lesson 2 : Introduction to ML

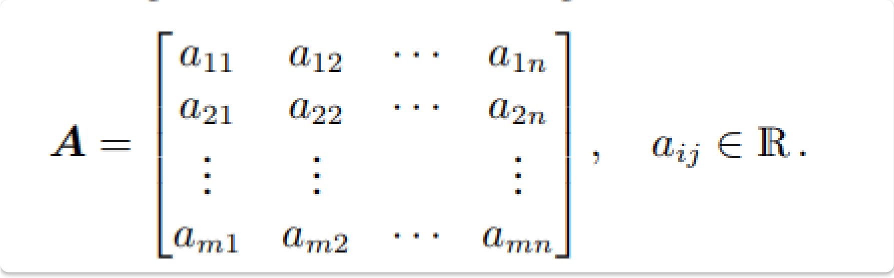
Lesson 3 :Intro to zkML / Use cases

Lesson 4 :EZKL

Today's topics

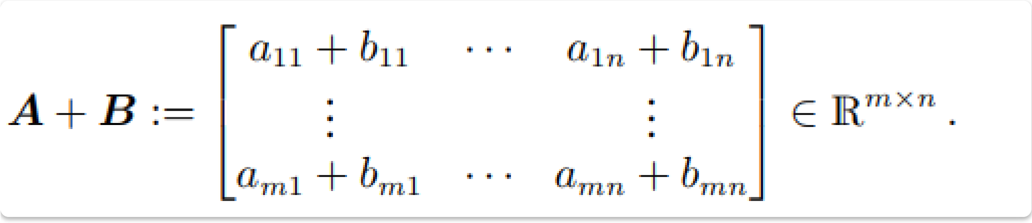
- Maths for ML
- Machine Learning Introduction
- (Un) Supervised Learning
- Neural network introduction
- Components
  - Nodes
  - Weights
  - Activation functions
- Hardware

> ### Matrices
>
> A matrix is an $m.n$ tuple of elements in a rectangular scheme of m rows and n columns



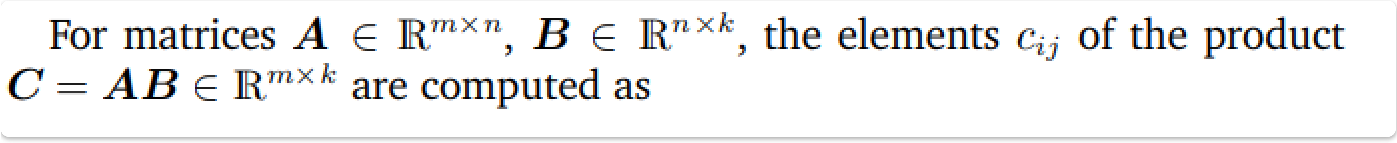
$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}, \quad a_{ij} \in \mathbb{R}.$$

Adding matrices

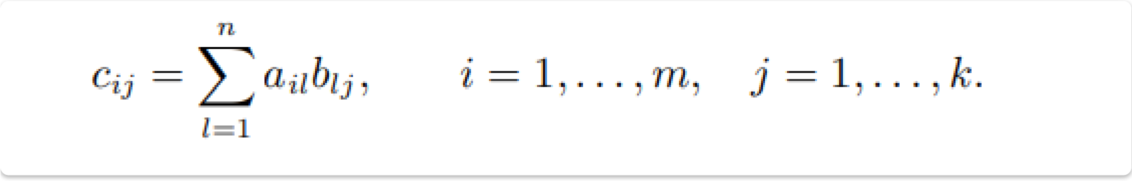


$$\boldsymbol{A} + \boldsymbol{B} := \begin{bmatrix} a_{11} + b_{11} & \cdots & a_{1n} + b_{1n} \\ \vdots & & \vdots \\ a_{m1} + b_{m1} & \cdots & a_{mn} + b_{mn} \end{bmatrix} \in \mathbb{R}^{m \times n}.$$

Multiplication



For matrices $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, $\boldsymbol{B} \in \mathbb{R}^{n \times k}$, the elements c_{ij} of the product $\boldsymbol{C} = \boldsymbol{AB} \in \mathbb{R}^{m \times k}$ are computed as

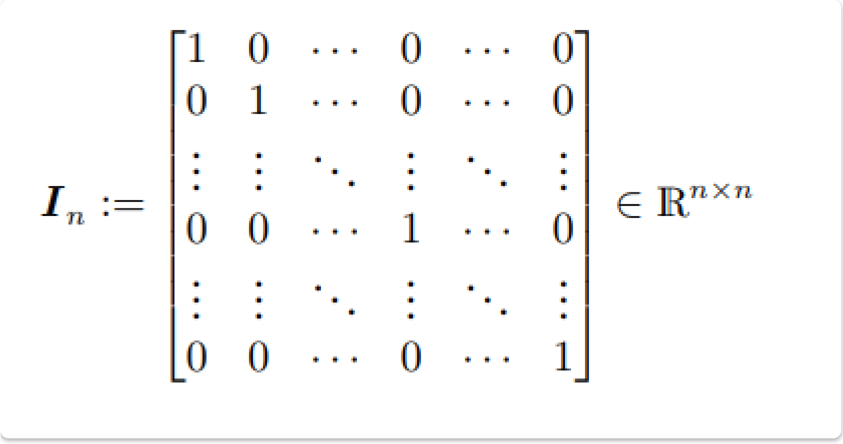


$$c_{ij} = \sum_{l=1}^{n} a_{il} b_{lj}, \qquad i = 1, \ldots, m, \quad j = 1, \ldots, k.$$

This means, to compute element c_{ij} we multiply the elements of the ith row of A with the jth column of B and sum them up.

We have to make sure that the dimensions of the matrices match, for example an n x k matrix A an be multiplied by a k x m matrix B

Identity Matrix



$$\boldsymbol{I}_n := \begin{bmatrix} 1 & 0 & \cdots & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & \cdots & 1 \end{bmatrix} \in \mathbb{R}^{n \times n}$$

The identity matrix has 1 on the diagonal and 0 everywhere else

Inverse Matrix

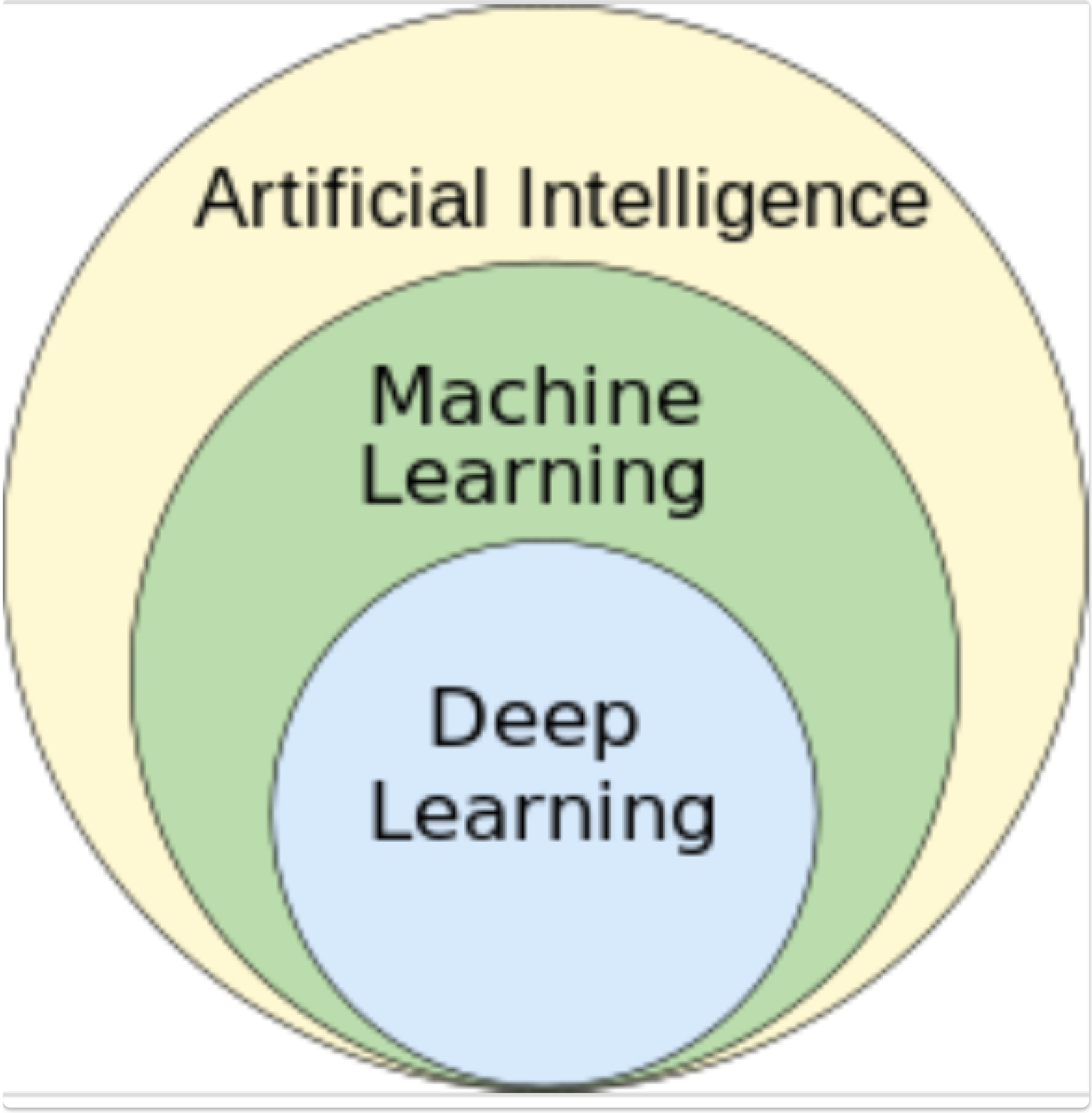
If A B = I_n then B is the inverse of A

Transpose Matrix

If we write the columns of A as the rows of B , then B is the transpose of A

Machine Learning - Introduction

Machine learning (ML) is a field of study in artificial intelligence concerned with the development and study of statistical algorithms that can learn from data and generalize to unseen data, and thus perform tasks without explicit instructions



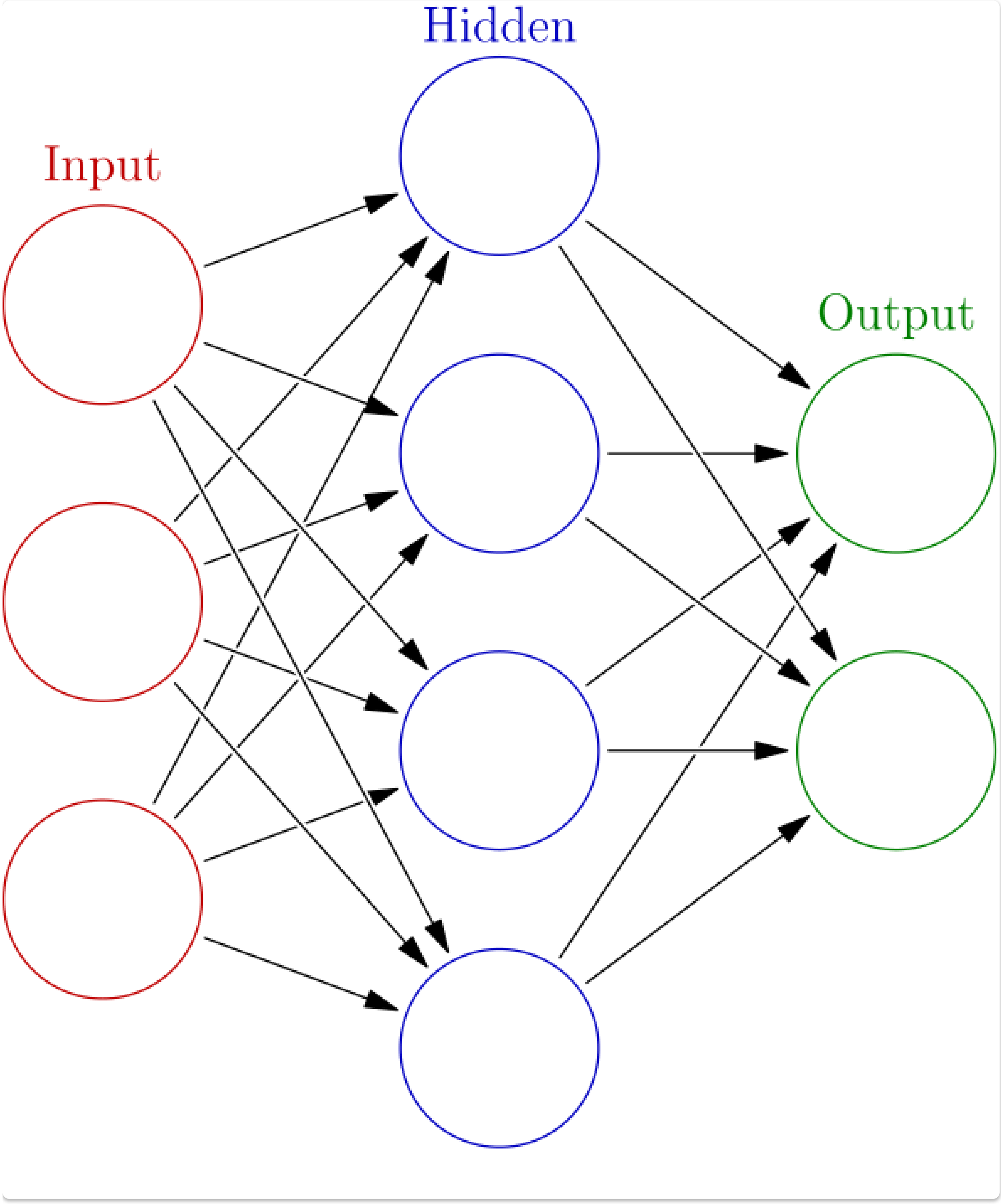
> ### Major Approaches
>
> - Supervised Learning
> - Unsupervised Learning
> - Reinforcement Learning
>
> #### Supervised Learning
>
> Supervised learning methods provide training patterns together with appropriate desired outputs
>
> The training set consists of input patterns with correct results so that the network can receive a precise error vector1 can be re- turned.
>
> #### Unsupervised Learning
>
> No labels are given to the learning algorithm, leaving it on its own to find structure in its input. Unsupervised learning can be a goal in itself (discovering hidden patterns in data) or a means towards an end (feature learning).

> **Reinforcement learning**
>
> Reinforcement learning methods provide feedback to the network, whether it behaves well or badly.
>
> In reinforcement learning the network receives a logical or a real value after completion of a sequence, which defines whether the result is right or wrong. Intuitively it is clear that this procedure should be more effective than unsupervised learning since the network receives specific criteria for problem-solving.

> **Process**
>
> The process of training and using a machine learning model has two main parts.
>
> - Machine learning training is the process of using an ML algorithm to build a model. It typically involves using a training dataset and a deep learning framework like TensorFlow.
> - Machine learning inference is the process of using a pre-trained ML algorithm to make predictions.

> **Models**
>
> In the field of machine learning, a model is a mathematical formula which, after being "trained" on a given dataset, can be used to make predictions or classifications on new data. During training, a learning algorithm iteratively adjusts the model's internal parameters to minimize errors in its predictions.

Neural Networks

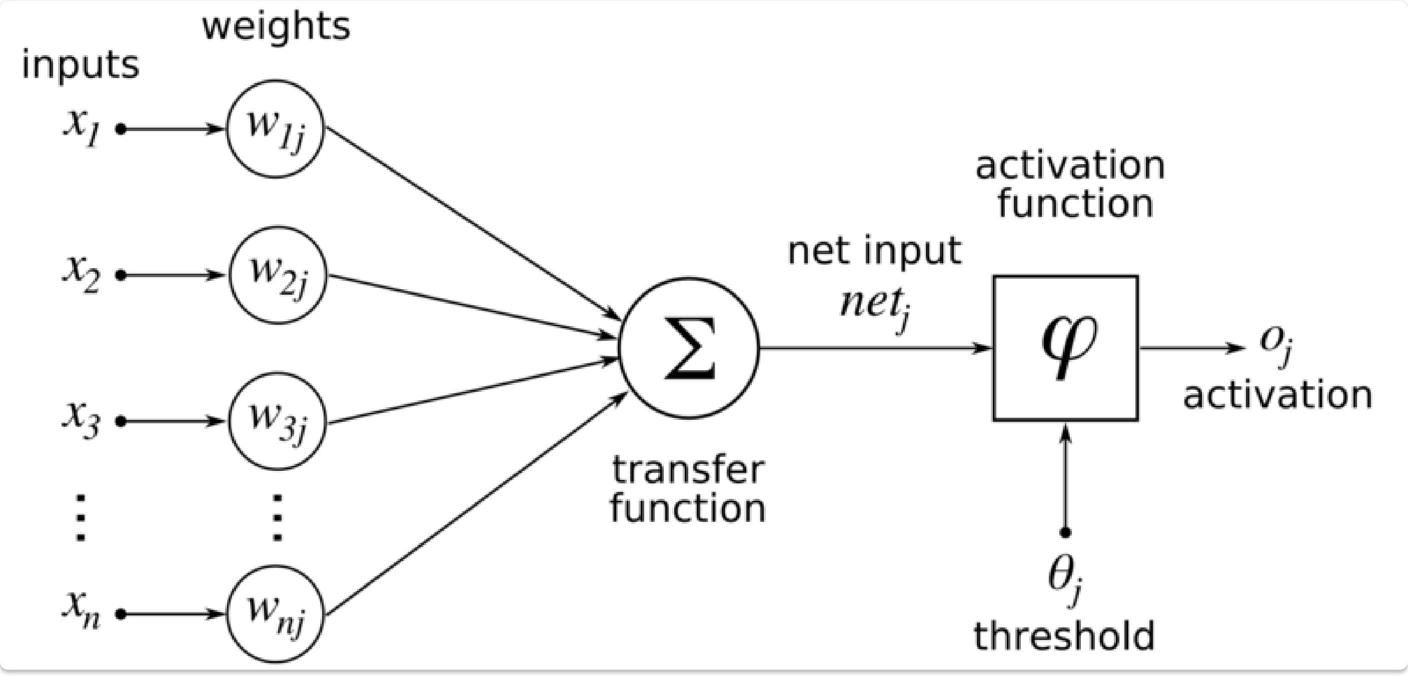


In common neural network implementations, the signal at a connection between artificial neurons is a real number, and the output of each artificial neuron is computed by some non-linear function of the sum of its inputs.

> Artificial neurons typically have a weight that adjusts as learning proceeds.

Components of a Neural Network

Neurons get activated if the network input exceeds their threshold value
The activation function determines the activation of a neuron dependent on network input and threshold value



Transfer function / Activation function

The transfer function translates the input signals to output signals. Four types of transfer functions are commonly used, Unit step (threshold), sigmoid, piecewise linear, and Gaussian.
The output is set at one of two levels, depending on whether the total input is greater than or less than some threshold value.

> ### Sigmoid Function
>
> See Description
>
> A Sigmoid function is a mathematical function which has a characteristic S-shaped curve.
>
> All sigmoid functions have the property that they map the entire number line into a small range such as between 0 and 1, or -1 and 1, so one use of a sigmoid function is to convert a real value into one that can be interpreted as a probability.
>
> ### Softmax Function
>
> See Description
>
> The softmax function is a function that turns a vector of K real values into a vector of K real values that sum to 1.
>
> The input values can be positive, negative, zero, or greater than one, but the softmax transforms them into values between 0 and 1, so that they can be interpreted as probabilities.

Training

Back Propagation

This is an optimisation algorithm, designed to minimise a `loss function`

Loss Function

Also known as a cost function or error function. It gives us an indication of how close our model is to producing the outputs that we want.

The loss function maps values of one or more variables onto a real number representing some "cost" associated with those values. For back propagation, the loss function calculates the difference between the network output and its expected output, after a training example has propagated through the network.

Our training process then is

- Entering the input pattern (activation of input neurons),
- Forward propagation of the input by the network, generation of the output,
- Comparing the output with the desired output giving the error function.

Convolutional Neural Networks

Convolution is a mathematical operation that combines two functions to produce a third function that represents how one function modifies the other. In signal processing, convolution is often used to manipulate and analyse signals.
A tensor is a multidimensional array.

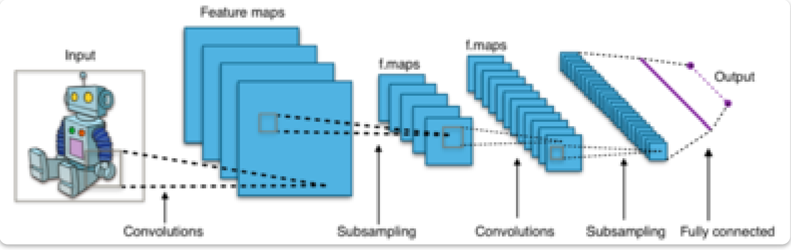
A CNN consists of an input layer, hidden layers and an output layer.

In a CNN, the input is a tensor with shape:

(number of inputs) × (input height) × (input width) × (input channels)

After passing through a convolutional layer, the image becomes abstracted to a feature map, also called an activation map, with shape:

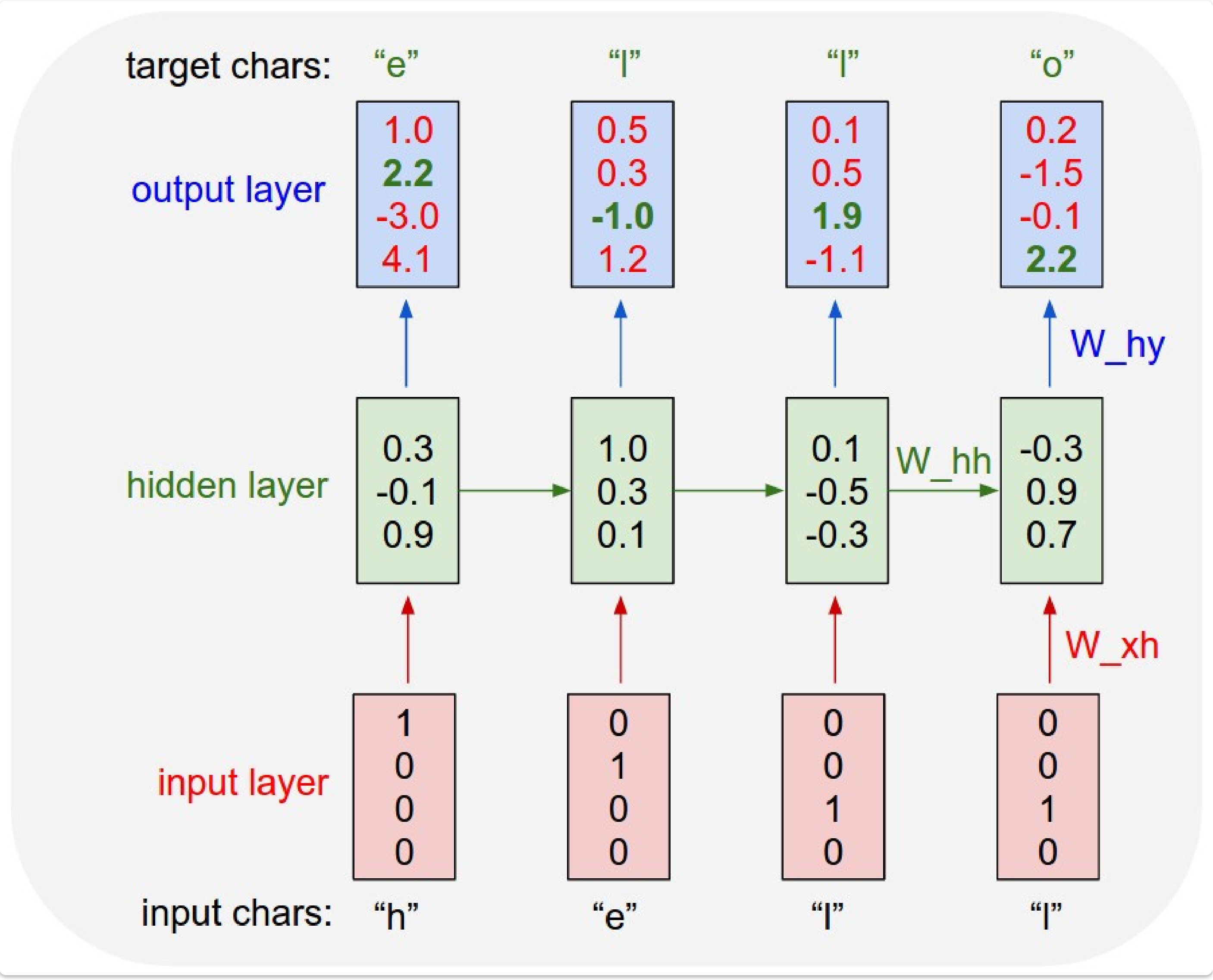
(number of inputs) × (feature map height) × (feature map width) × (feature map channels).



> Sub-sampling is a technique that has been devised to reduce the reliance of precise positioning within feature maps that are produced by convolutional layers within a CNN. To reduce the reliance on the exact positioning of features within networks, the reduction of spatial resolution is undertaken.

Recurrent Neural Networks

See Blog

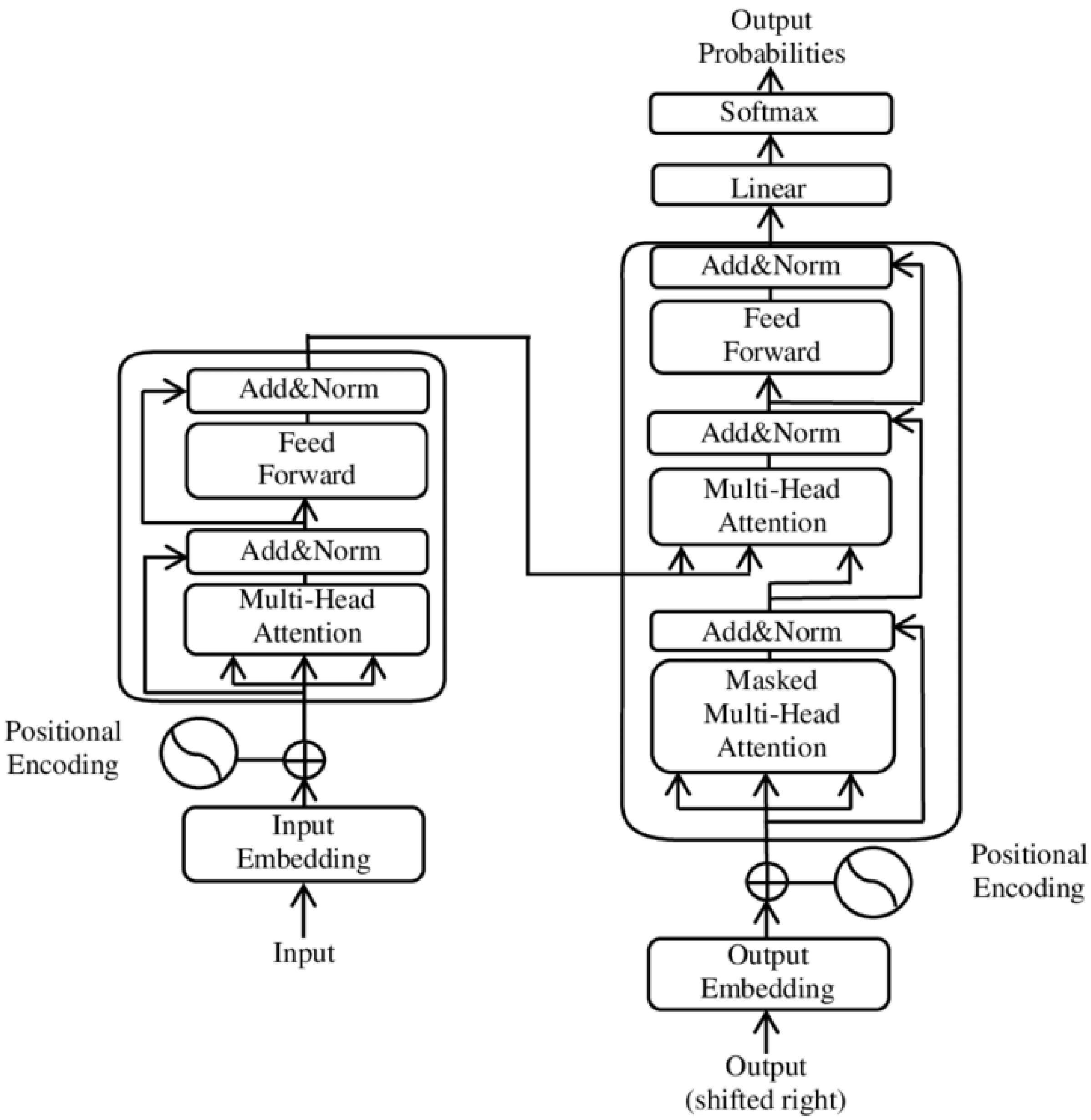


An example RNN with 4-dimensional input and output layers, and a hidden layer of 3 units (neurons). This diagram shows the activations in the forward pass when the RNN is fed the characters "hell" as input. The output layer contains confidences the RNN assigns for the next character (vocabulary is "h,e,l,o"); We want

> the green numbers to be high and red numbers to be low.
>
> Example Shakespeare
>
> ```
> KING LEAR:
> O, if you were a feeble sight, the
> courtesy of your law,
> Your sight and several breath, will wear
> the gods
> With his heads, and my hands are wonder'd
> at the deeds,
> So drop upon your lordship's head, and
> your opinion
> Shall be against your honour.
> ```

Transformers

A transformer is a deep learning architecture based on an attention mechanism, proposed in a 2017 paper "Attention Is All You Need"
It has no recurrent units, and thus requires less training time than RNNs and later variations have formed the basis for training Large Language Models on large datasets



Hardware for AI

See summary article
ML and particularly Deep Learning benefit from parallelisation when training (or running inference).
Although it is possible to use CPUs for this, it is more suited to different hardware architectures such as GPUs or FPGAs

GPU

GPUs are specialised hardware components that can perform numerous simple operations simultaneously. GPUs and CPUs share a similar structure—both employing spatial architectures—but otherwise vary greatly.

FPGA

FPGA is specialized hardware that users can configure after manufacturing. It includes:

- An array of programmable logic blocks
- A hierarchy of configurable interconnections

This hierarchy enables inter-wiring blocks in different configurations. Users can write code in

> a hardware description language (HDL) like VHDL or Verilog, and the program determines the connections and how to use digital components to implement them.
>
> FPGAs can support a vast amount of multiply and accumulate operations. This ability enables FPGAs to implement parallel circuits. However, HDL is a piece of code that defines hardware components like counters and registers—it is not a programming language. It makes some aspects very difficult, such as converting your Python library to FPGA code.