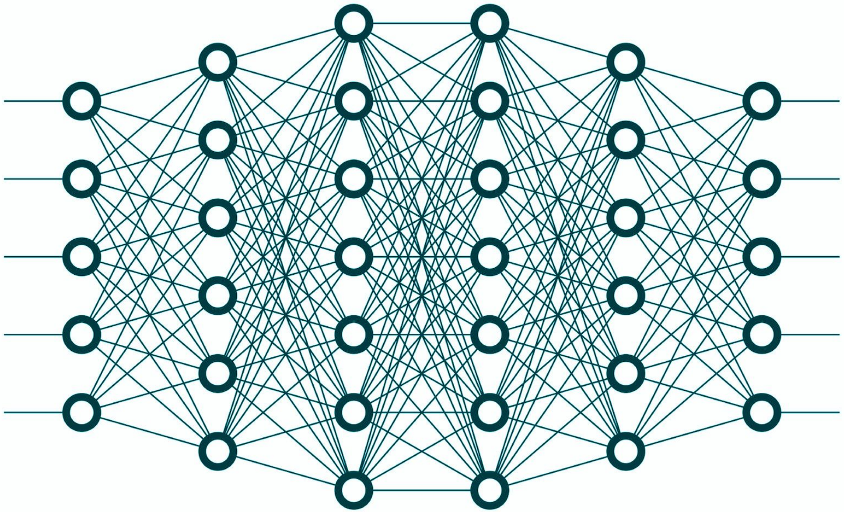
Deep learning

Lecture 1 - Intro to Neural Networks

12.05.2025

Toker Gilat



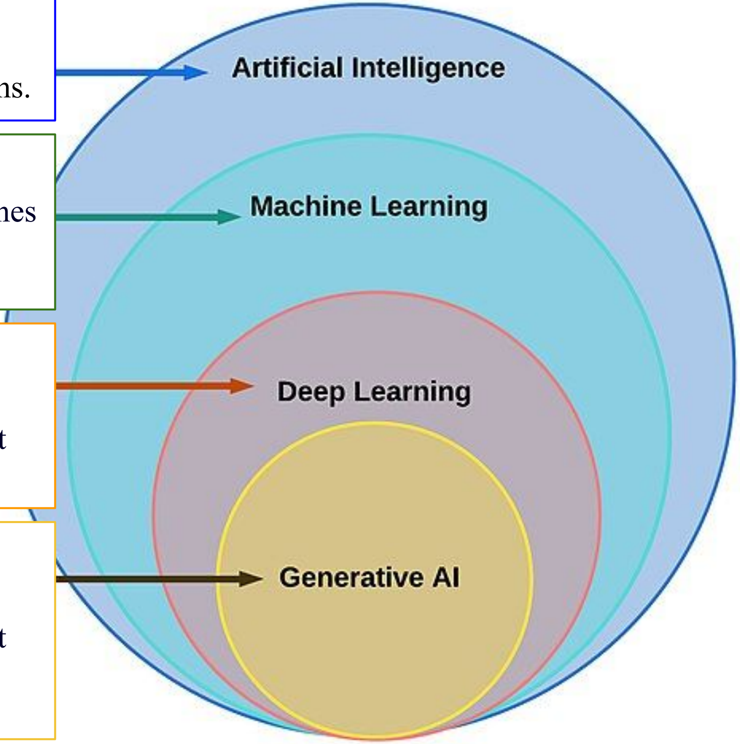
What is deep learning?

Artificial Intelligence - AI involves techniques that equip computers to emulate human behavior, enabling them to learn, make decisions, recognize patterns, and solve complex problems.

Machine Learning - ML is a subset of AI, uses advanced algorithms to detect patterns in large data sets, allowing machines to learn and adapt. ML algorithms use **supervised or unsupervised learning methods.**

Deep Learning - DL is a subset of ML which uses neural networks for in-depth data processing and analytical tasks. DL leverages multiple layers of artificial neural networks to extract high-level features, simulating the human brains perceive.

Generative AI - DL is a subset of ML which uses neural networks for in-depth data processing and analytical tasks. DL leverages multiple layers of artificial neural networks to extract high-level features from raw input data, simulating the way human brains perceive and understand the world.



https://commons.wikimedia.org/wiki/File:Unraveling_AI_Complexity_-_A_Comparative_View_of_AI,_Machine_Learning,_Deep_Learning,_and_Generative_AI.jpg

Main Differences between ML and DL :

Machine Learning (ML):

- Utilizes statistical algorithms to find patterns in data.
- Works well with smaller datasets.
- Faster to train compared to DL models.
- Requires manual feature extraction (e.g., from images or text).
- Easier to interpret and less complex.
- Can run efficiently on a CPU.

Deep Learning (DL):

- Uses artificial neural networks to learn patterns automatically.
- Requires large datasets for effective training.
- Slower to train, especially with complex models.
- Performs end-to-end feature extraction and learning.
- More complex and less interpretable (black-box nature).
- Needs a high-performance GPU for efficient computation.

Why Now?

Deep Learning is not new, but its growth in the past decade has been exponential.
Key Factors Driving Deep Learning Success:

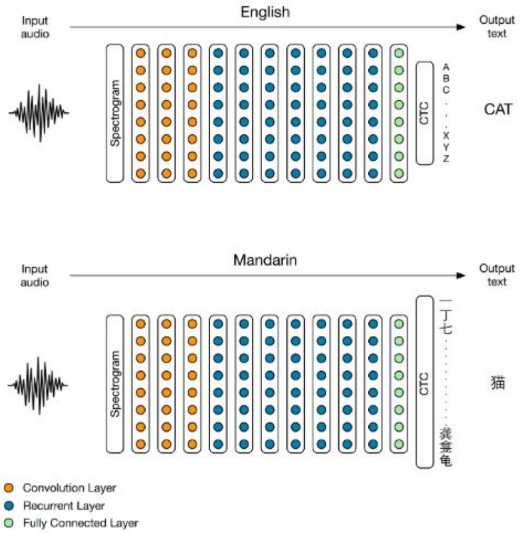
1. **Explosion of Data:** Now, Massive amounts of data are generated daily (e.g., social media, IoT devices, and web activity).
2. **Advancements in Computing:**
   - Hardware Revolution: GPUs (NVIDIA) and TPUs (Google) optimized for parallel processing of large-scale computations.
   - Distributed Computing: Cloud platforms (e.g., AWS, Google Cloud) democratize access to powerful hardware.
3. **Pre-Trained Models:** Transfer learning allows leveraging pre-trained models, drastically reducing time and resources for new tasks.
   Examples: ResNet (image classification), BERT (NLP).
4. **Open-Source Ecosystem:** Tools like TensorFlow, PyTorch, and Hugging Face made DL accessible.

DL Today is Everywhere

To better understand its impact, let's walk through the main fields where DL is widely used and driving rapid progress:

DL Today: Speech-to-Text

Deep learning models have revolutionized **automatic speech recognition (ASR)**, enabling machines to understand spoken language with near-human accuracy.



[Baidu 2014]

Taken from: https://m2dsupsdlclass.github.io/lectures-labs/

DL Today is Everywhere

To better understand its impact, let's walk through the main fields where DL is widely used and driving rapid progress:

DL Today: Vision

Image classification [Krizhevsky 2012] — AlexNet's using CNNs, outperformed SOTA.

Handwritten character recognition [Ciresan et al. 2013] — Deep neural nets proved successful in reading complex visual symbols , paving the way for handwriting input systems.

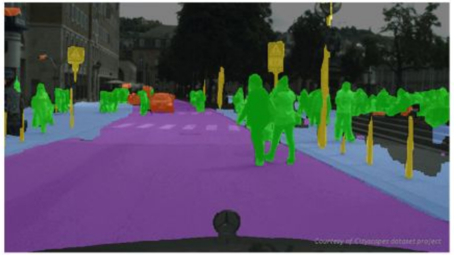
Object detection [Faster R-CNN, Ren 2015] — Integrated **region** proposal networks with CNNs to detect and localize multiple objects within an image (crucial for robotics, autonomous driving).

Semantic segmentation [NVIDIA dev blog] — NVIDIA's hierarchical multi-scale attention approach enhances pixel-wise classification (e.g., road, pedestrian, sky), enabling detailed scene understanding.

[Krizhevsky 2012]

[Ciresan et al. 2013]

[Faster R-CNN - Ren 2015]



[NVIDIA dev blog]

Taken from: https://m2dsupsdlclass.github.io/lectures-labs/

DL Today is Everywhere

To better understand its impact, let's walk through the main fields where DL is widely used and driving rapid progress:

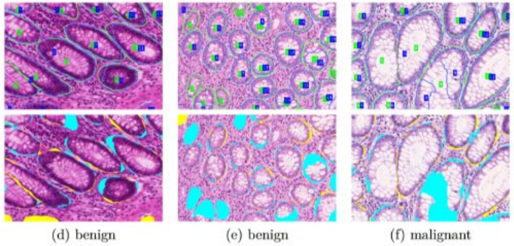
Facial Landmark Detection [CUHK 2014] proposed a deep multi-task learning framework for facial landmark detection (eyes, nose, lips), improving robustness against occlusions and pose variations.

Face Recognition FaceNet by Schroff et al. (2015) presented a system that maps face images to a compact Euclidean space, enabling robust recognition invariant to pose, lighting, and age.

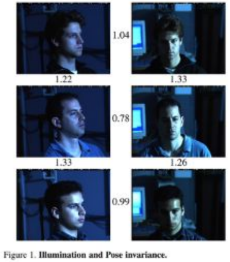
CheXNet by Rajpurkar et al. (2017) introduced a 121-layer DenseNet trained on the ChestX-ray14 dataset, achieving radiologist-level performance in pneumonia detection from chest X-rays.

DL Today: Vision

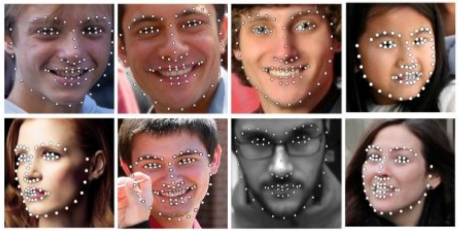
[Stanford 2017]



[Nvidia Dev Blog 2017]



[FaceNet - Google 2015]



[Facial landmark detection CUHK 2014]

Taken from: https://m2dsupsdlclass.github.io/lectures-labs/

DL Today is Everywhere

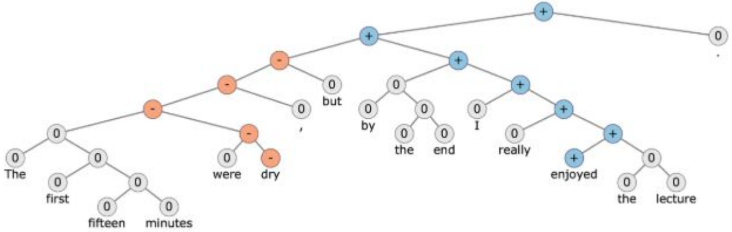
To better understand its impact, let’s walk through the main fields where DL is widely used and driving rapid progress:

DL Today: NLP

Socher et al. developed RNTNs (2015) to capture compositional semantics in sentences, achieving state-of-the-art results in sentiment analysis by modeling the hierarchical structure of language.

Google's Neural Machine Translation (GNMT), 2016 introduced an end-to-end system using deep LSTM networks with attention mechanisms, significantly improving translation quality.

[Google Translate System - 2016]



[Socher 2015]

Taken from: https://m2dsupsdlclass.github.io/lectures-labs/

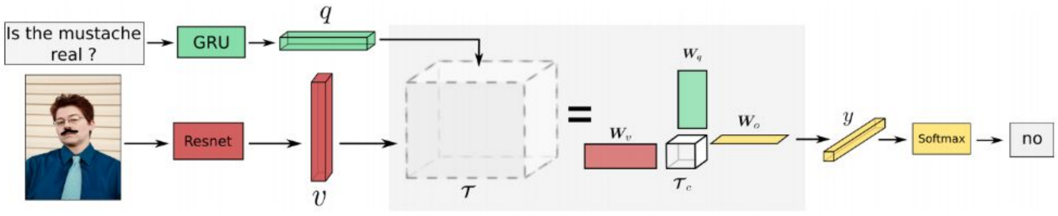
DL Today is Everywhere

To better understand its impact, let’s walk through the main fields where DL is widely used and driving rapid progress:

DL Today: Vision + NLP

Image Captioning [Karpathy et al., 2015] - Aligns **image regions** with **corresponding words in sentences** using a combination of CNNs and RNNs). The model learns to generate natural language descriptions for images by grounding textual phrases in specific visual regions

Visual Question Answering (VQA) [Mutan 2017]- technique for combining visual and textual information in VQA. It combines visual features (from CNNs) and language features (from RNNs) using a bilinear pooling mechanism



[VQA - Mutan 2017]

[Karpathy 2015]

Taken from: https://m2dsupsdlclass.github.io/lectures-labs/

DL Today is Everywhere

To better understand its impact, let's walk through the main fields where DL is widely used and driving rapid progress:

DL Today: Generative models

StackGAN (ICCV 2017) Introduced a two-stage GAN architecture for text-to-image synthesis. Stage-I GAN generates low-resolution (64×64) images capturing basic shapes and colors from text descriptions. Stage-II GAN refines these into high-resolution (256×256) images.

ProGAN (NVIDIA 2017) - Proposed a progressive training methodology for GANs, starting from low-resolution images (4×4) and incrementally adding layers to both generator and discriminator to handle higher resolutions. This approach stabilizes training and enables the generation of high-quality images up to 1024×1024 resolution.

Sampled celebrities [Nvidia 2017]

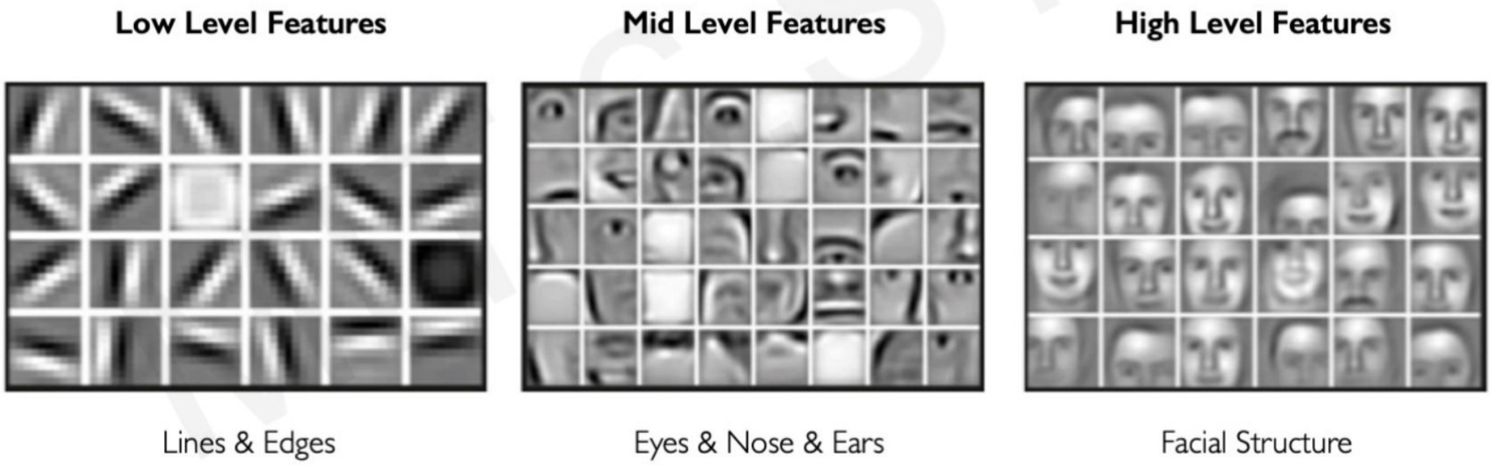
StackGAN v2 [Zhang 2017]

Taken from: https://m2dsupsdlclass.github.io/lectures-labs/

What is deep learning?

Key Characteristics:

1. **Automatic Feature Learning:**
   DL models extract features without human intervention.
   Example: In image recognition, DL identifies edges, shapes, and objects from pixels.



What is deep learning?

Key Characteristics:

1. **Automatic Feature Learning:**
   DL models extract features without human intervention.
   Example: In image recognition, DL identifies edges, shapes, and objects from pixels.

2. **Large-Scale Models:**
   - Can handle vast amounts of data and complex patterns.
   - Often requires GPUs/TPUs for efficient training.

What is deep learning?

Key Characteristics:

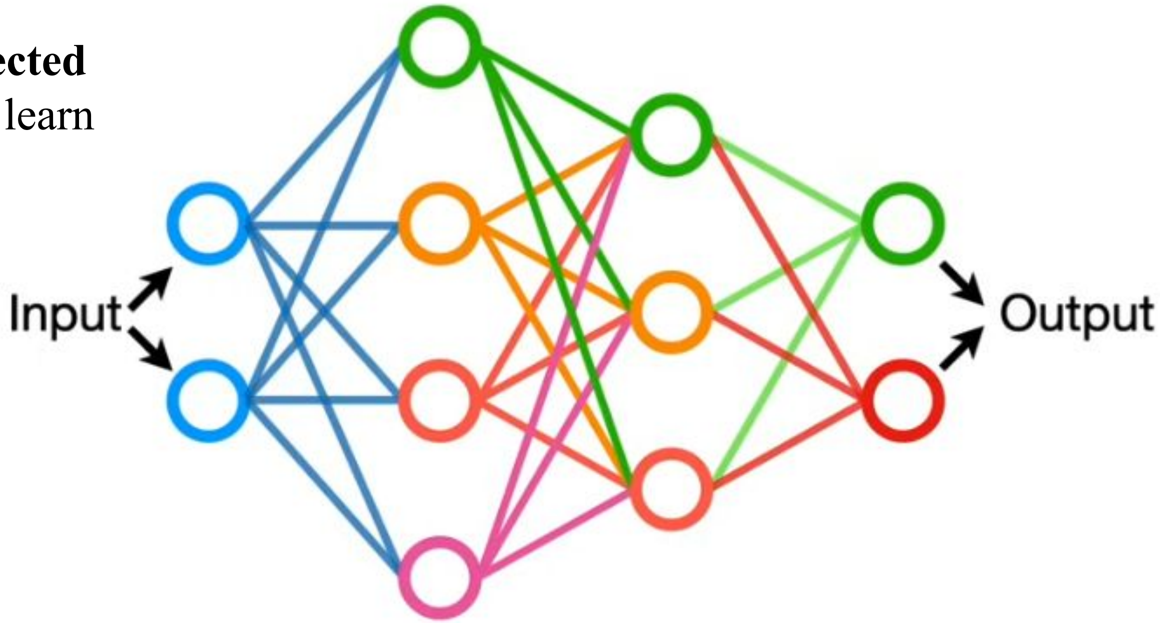
1. **Automatic Feature Learning:**
   DL models extract features without human intervention.
   Example: In image recognition, DL identifies edges, shapes, and objects from pixels.

2. **Large-Scale Models:**
   - Can handle vast amounts of data and complex patterns.
   - Often requires GPUs/TPUs for efficient training.

3. **Multi-Layered Neural Networks:**
   - Each layer extracts different levels of abstraction:
   - Early layers detect simple patterns (e.g., edges in images).
   - Deeper layers capture more complex structures (e.g., faces).

What is Neural Network?

A neural network is a machine learning model inspired by the structure and function of the human brain.
It consists of **layers of interconnected nodes (neurons)** that process and learn from data.

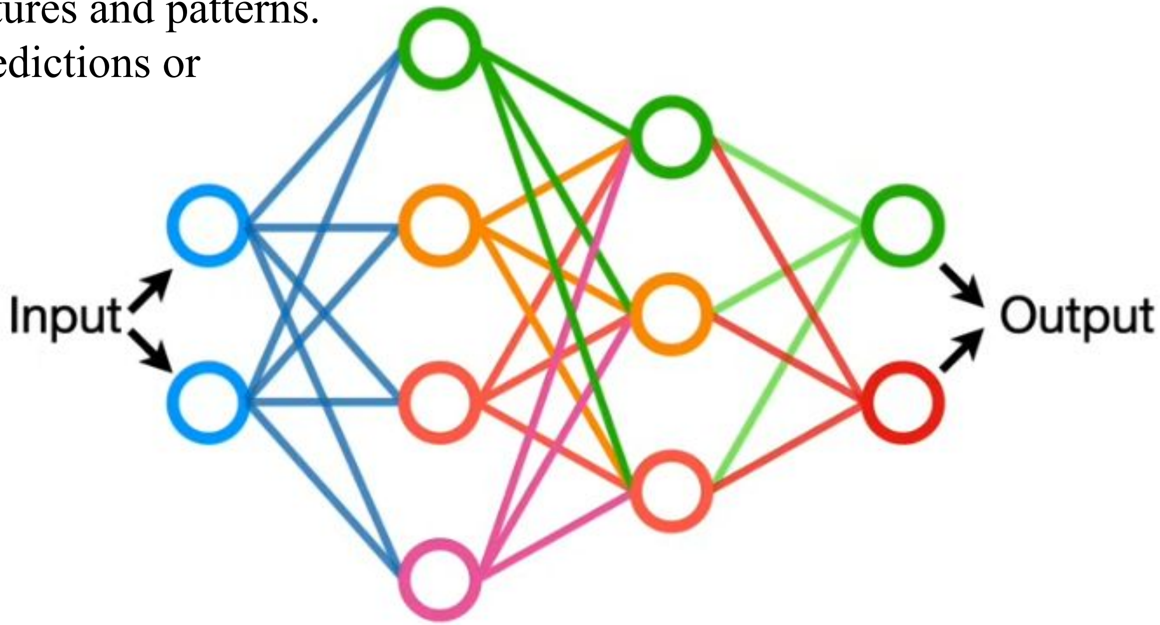


https://www.youtube.com/watch?v=CqOfi41LfDw

What is Neural Network?

Key Components:

- **Input Layer:** Receives the data.
- **Hidden Layers:** Extract features and patterns.
- **Output Layer:** Produces predictions or classifications.



How does deep learning work?

Deep learning works by using artificial neural networks to learn representations from data. Neural networks are composed of **layers** of **interconnected nodes (neurons)**, and each node is responsible for learning a specific feature of the data.

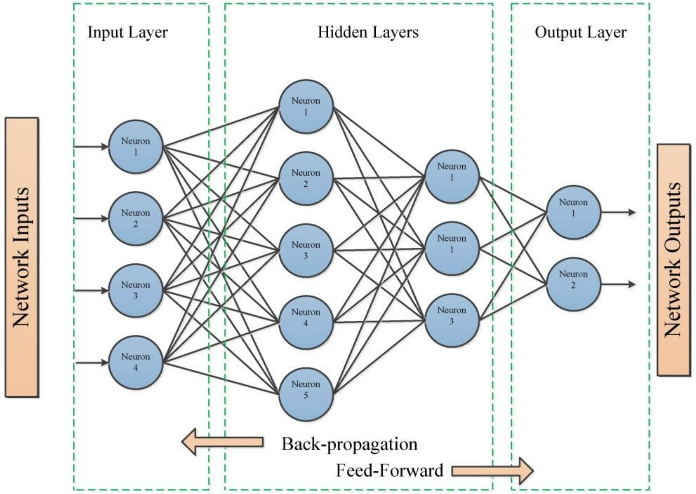
As the network learns, the weights on the connections between nodes are adjusted to minimize the prediction error, so that the network can better classify the data..

Once a neural network has been trained, it can be used to make predictions on new, unseen data by applying the learned transformations.

https://cloud.google.com/discover/what-is-deep-learning

Neural Network Workflow

Forward Propagation: Data flows from the input layer through the hidden layers to the output. Each neuron applies a weight and a bias, followed by an activation function.
Loss Calculation: Compares the predicted output with the true value using a loss function (e.g., Mean Squared Error for regression).
Backward Propagation: Adjusts weights and biases by minimizing the loss using gradient descent.



Components of a Neural Network

Neural networks are built from several key components:

1. **Neurons:** Each neuron receives inputs, applies weights and biases, and produces an output.
   Mathematical Representation:

$$z = \sum(w \cdot x) + b$$

   where z is the weighted sum, w are weights, x are inputs, and b is the bias.

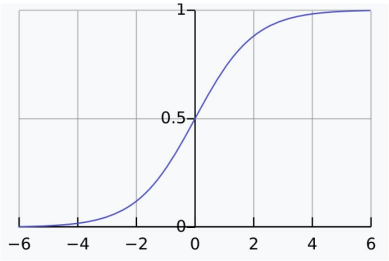
Components of a Neural Network

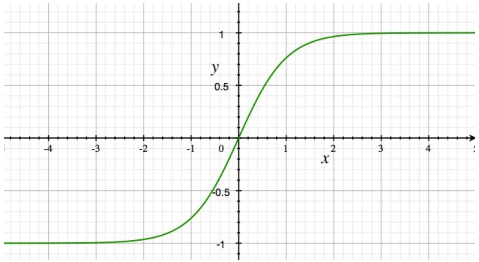
Neural networks are built from several key components:

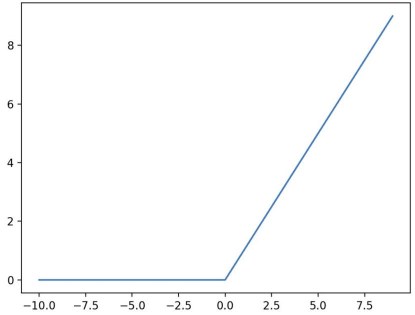
2. Activation Functions: Activation functions introduce non-linearity to the model, enabling it to learn complex patterns.
Without activation functions, the network would behave like a linear regression model, regardless of depth.

Common Activation Functions:

- Sigmoid: $\sigma(x) = \frac{1}{1+e^{-x}}$
- Tanh $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$
- ReLU $f(x) = \max(0, x)$

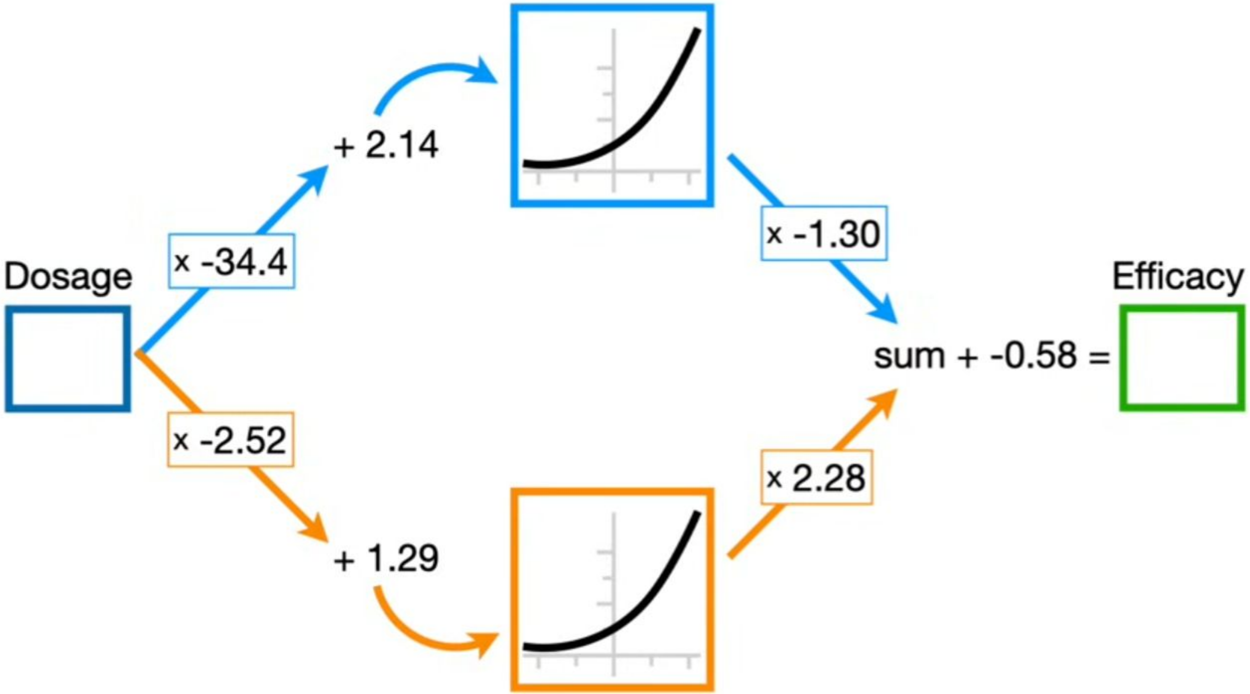






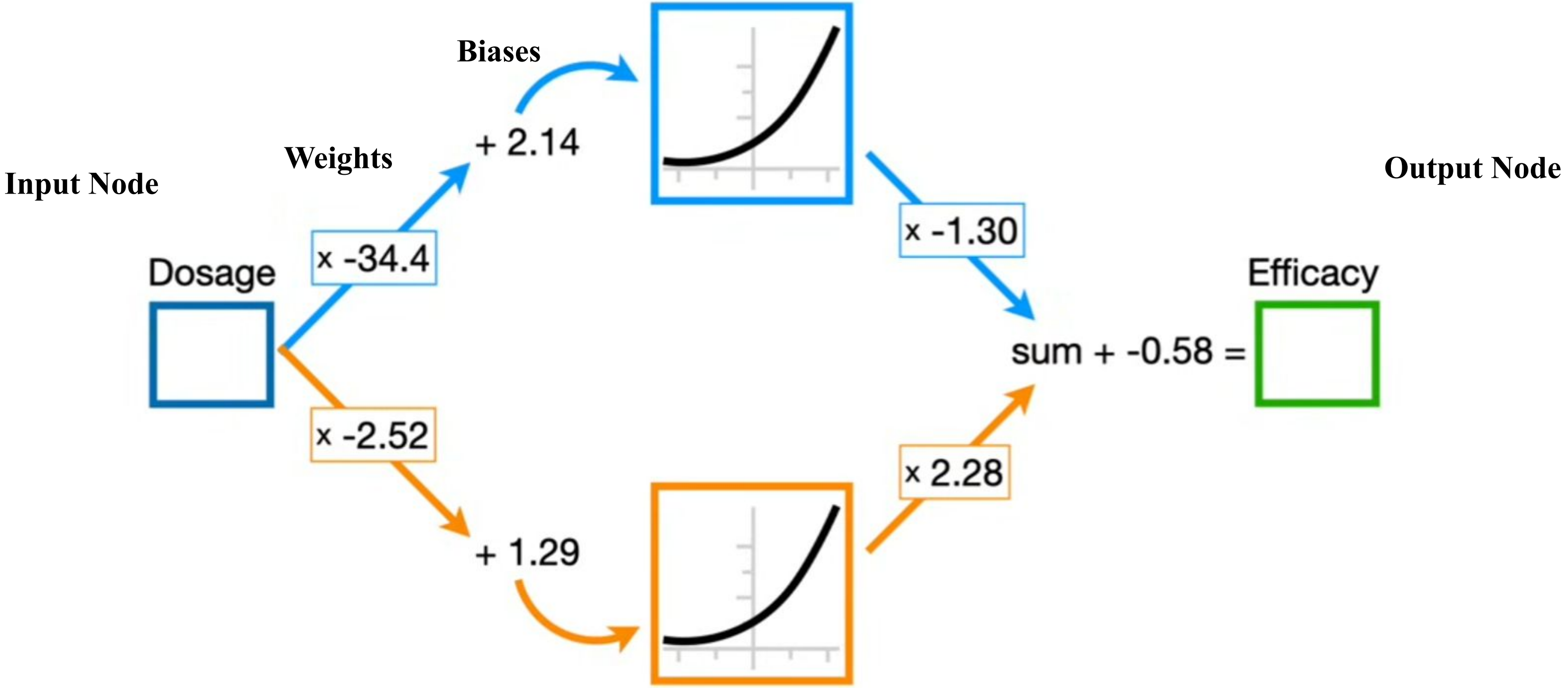
What is deep learning?

3. Weights and Biases: Weights determine the influence of a given input, biases help adjust the model to better fit the data.



What is deep learning?

3. Weights and Biases: Weights determine the influence of a given input, biases help adjust the model to better fit the data.



Components of a Neural Network

Neural networks are built from several key components:

4. loss functions: Quantify how far predictions are from actual values.
Guide the optimization process.
Common loss functions: Mean Squared Error (MSE), Cross-Entropy Loss

5. optimization algorithms: The optimizer's job is to determine which combination of the neural network's weights and biases will give it the best chance to generate accurate predictions.

What is Backpropagation?

Backpropagation is a key algorithm for **training** neural networks by **optimizing weights.** It **computes the gradient of the loss function** with respect to each weight **by applying the chain rule.**

How It Works:
Forward Pass: Inputs are passed through the network to compute the output (prediction). Loss is calculated using a chosen loss function (e.g., Mean Squared Error or Cross-Entropy).
Backward Pass: Gradients of the loss with respect to the weights are computed using the chain rule. These gradients are used to adjust the weights to minimize the loss.

Backpropagation is Important because it enables efficient optimization of deep networks. Allows the network to learn patterns from data by minimizing error.

Mathematical Foundation of Backpropagation?

Key Components:

1. Loss Function (L): Measures the difference between predicted output (y_{pred}) and actual output (y_{true}).
   Example:
   $$L = \frac{1}{2}(y_{true} - y_{pred})^2$$
2. Chain Rule: Gradients are computed using the chain rule, layer by layer, from the output layer backward to the input. For example:
   $$\frac{\partial L}{\partial w} = \frac{\partial L}{\partial y_{out}} \cdot \frac{\partial y_{out}}{\partial h} \cdot \frac{\partial h}{\partial w}$$
3. Weight Update Rule: After computing gradients via backpropagation, gradient descent is applied to update weights and minimize the loss:
   (η: Learning rate.)
   $$w \leftarrow w - \eta \frac{\partial L}{\partial w}$$

Hands on !

Open Google colab notebook - DL1_NeuralNetworks.ipynb