Unveiling the Mystery of Deep Learning: Past, Present, and Future

Dr. Elham Barezi,
AI Research scientist

Co-Sponsored by Rosen Center for Advanced Computing (RCAC), and IPAI

Spring 2025

PURDUE UNIVERSITY

Course Outline

1. History and Basics of DNN
   a. From traditional ML to DNN
2. Fundamental deep learning: from discriminative to generative
   a. CNN, RNN, Autoencoders, attention,
   b. Deep learning for Representation Learning and feature extraction
   c. Discriminative vs generative deep learning: VAE, GAN, Diffusion Models
3. Transformers Era
   a. self-attention, encoders, decoders, masking,
   b. Transformers for other modalities: text, image, video, speech,
4. LLMs in Practice
   a. Prompt Engineering Methods: COT, TOT, Self-Consistency, RAG, Agents,
   b. Fine-tuning Methods: instruct tuning, RLHF, Adapters like LORA,
5. Deep learning for different domains
6. AI safety and Governance

Course Outline-first session-March 5th 2025

1. History and Basics of DNN
    a. AI hypes and winters
    b. Deep learning from 1950s
    c. From single neurons to deep networks
    d. Deep learning challenges solved from 1950-present
        i. Model overfitting
        ii. Activation function saturation
        iii. Vanishing/exploding gradient
    e. Deep learning weaknesses

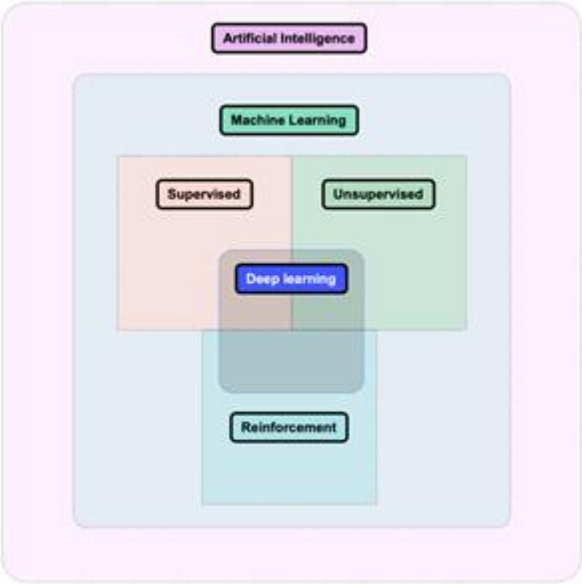
Some Definitions

Artificial intelligence (AI) is technology that enables computers and machines to simulate human learning, comprehension, problem solving, decision making, creativity and autonomy:

Machine learning, rule-based, symbolic AI, planning, Genetic Algorithms & Evolutionary Computation

Machine learning is a pathway to artificial intelligence, which uses algorithms to automatically learn insights and recognize patterns from data, make increasingly better decisions: supervised, unsupervised, reinforcement learning

Deep learning is an advanced method of machine learning. Deep learning models use large neural networks — networks that function like a human brain to logically analyze data — to learn complex patterns and make predictions.



Will there be another AI winter?

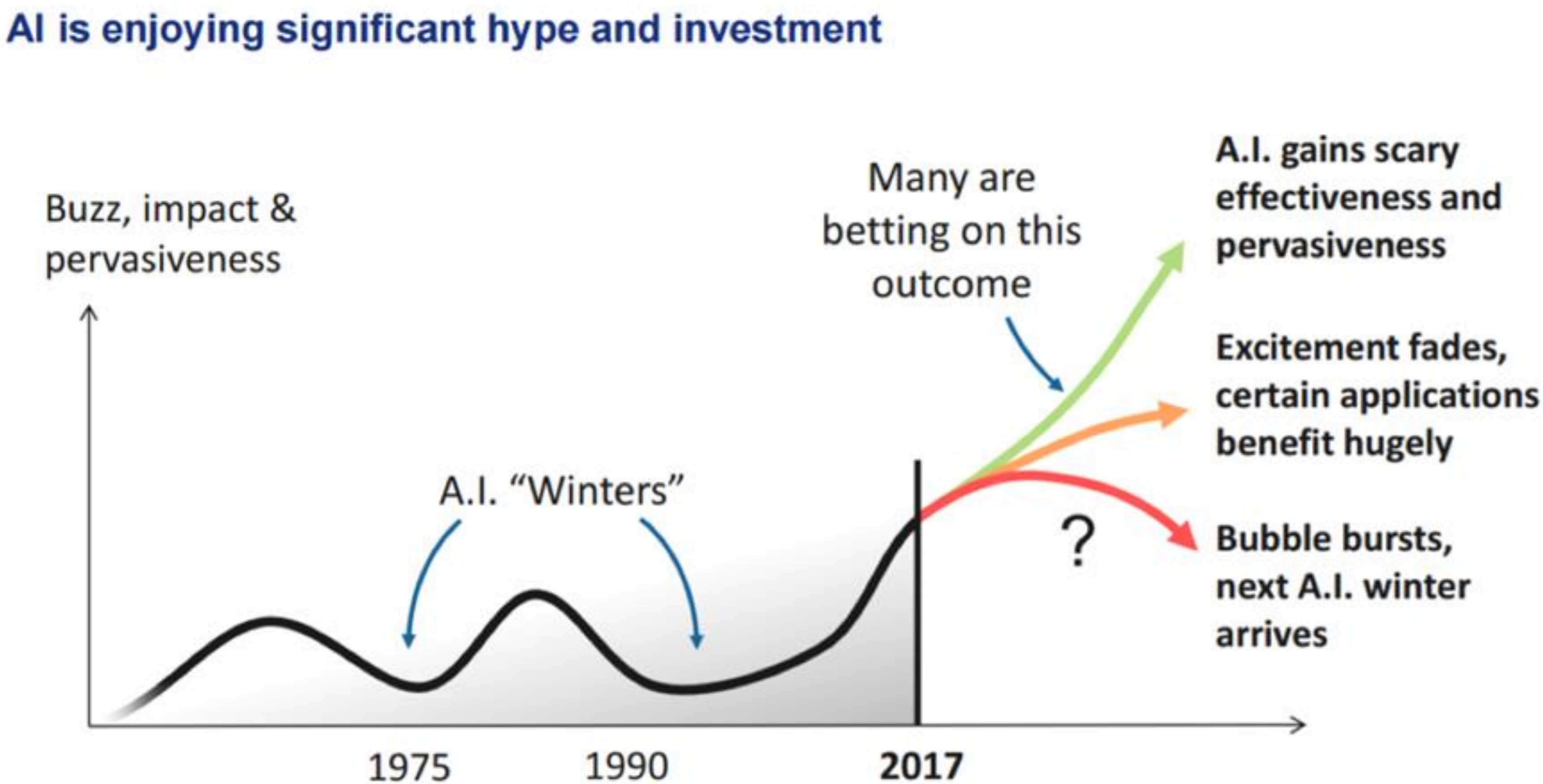


Photo taken from [9]

The beginning of AI research & First AI winter

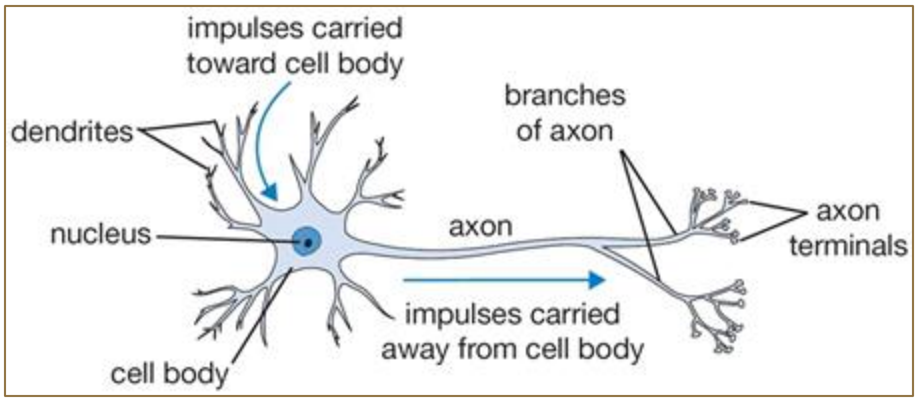
1956: Field of AI research founded at a workshop held in Dartmouth College (Beginning of AI research)
Many of the attendees predicted that a machine as intelligent as a human-being would exist in no more than a generation and they were given millions of dollars to make this vision come true.

1960: Massive investment in AI research
The Defense Advanced Research Projects Agency (now known as "DARPA") provided millions of dollars for AI research with almost no strings attached

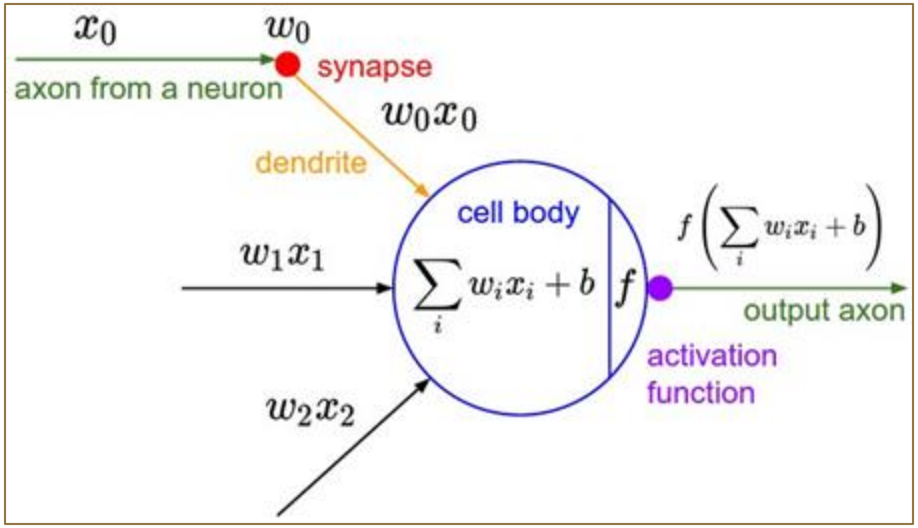
1969: DARPA started to be more conservative with their funds.
Only funded "mission-oriented direct research, rather than basic undirected research", so DARPA's money was directed at specific projects with identifiable goals (e.g. autonomous tanks and battle management systems)

1974: funding for AI projects was hard to find. (First AI winter)
Reports & study (Lighthill report, American Study Group) suggested that most AI research was unlikely to produce anything truly useful in the foreseeable future

Biological Neuron Structure



photos from: https://cs231n.github.io/neural-networks-1/



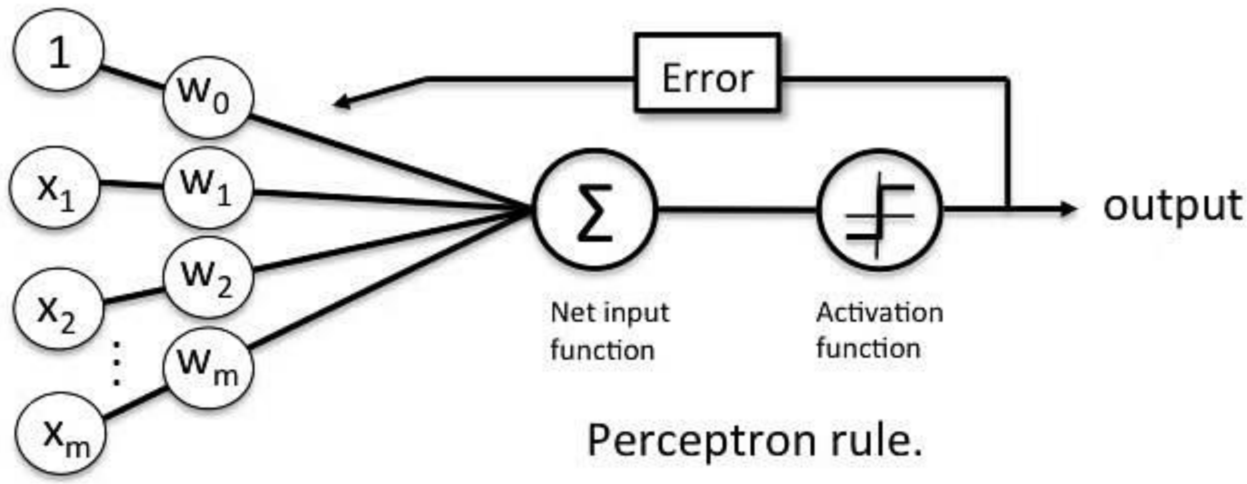
A drawing of a biological neuron (left) and its mathematical model (right)

PURDUE UNIVERSITY

Perceptron Model (1958)

If a misclassification occurs:

- If $y - \hat{y} = 1$, it **adds** a fraction of x_i to w_i, pushing the decision boundary in the correct direction.
- If $y - \hat{y} = -1$, it **subtracts** a fraction of x_i from w_i, moving in the opposite direction.



$$\hat{y} = f(w^T x + b) = \begin{cases} 1, & \text{if } w^T x + b \geq 0 \\ 0, & \text{if } w^T x + b < 0 \end{cases}$$

$$w_i \leftarrow w_i + \eta \cdot (y - \hat{y}) \cdot x_i$$

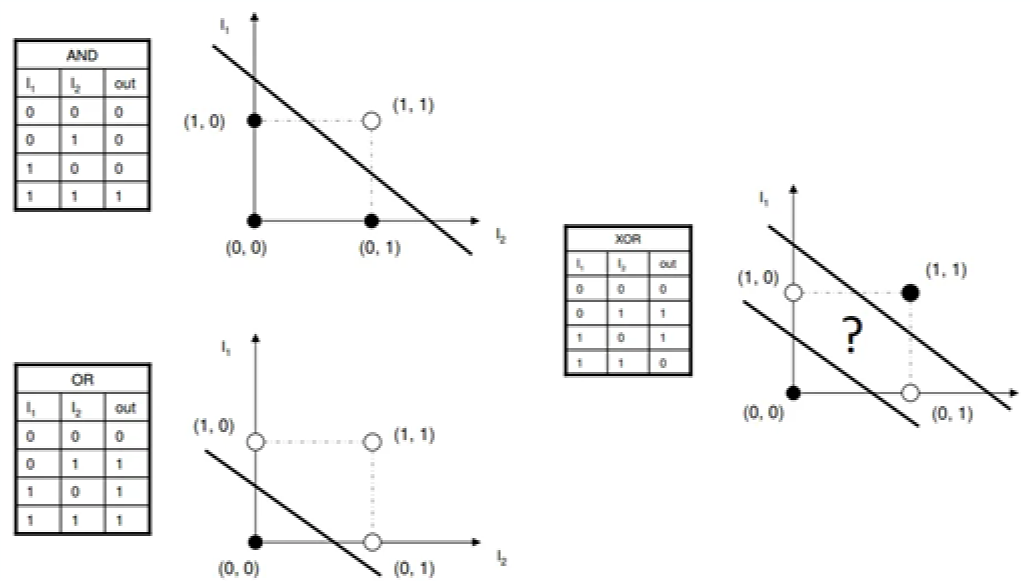
PURDUE UNIVERSITY

Photo from: https://medium.com/@musicaround/11-what-is-a-linear-classifier-logistic-regression-4eb44e2544b4

Perceptron weakness (1969)

First AI winter

Minsky et al 1969 proved that **single-layer Perceptrons cannot solve non-linearly separable problems**, such as the XOR function, which resulted in first AI winter until discovery of backpropagation at 1980s.



- **Rosenblatt** believed perceptrons could learn, recognize patterns, and eventually lead to AI.
- **Minsky & Papert (1969)** proved that single-layer perceptrons couldn’t solve non-linearly separable problems like XOR, limiting their power.
- This led to a decline in neural network research until **multi-layer perceptrons and backpropagation (1986)** revived deep learning.

Second AI winter

1980s: Development and adoption of a form of AI program called an "expert system"
The first commercial expert system was XCON, developed at Carnegie Mellon for Digital Equipment Corporation.

1985: Corporations around the world spent over a billion dollars on AI, most of it to in-house AI departments.
Enormous success of "expert systems". It was estimated to have saved the company 40 million dollars over just six years of operation.

1987: collapse of the market for specialized AI hardware, 3 years after Minsky and Schank's prediction.
Workstations by companies like Sun Microsystems offered a powerful alternative to LISP machines and later desktop computers built by Apple and IBM would also offer a simpler and more popular architecture to run LISP applications on.

1990s: Fail of the earliest successful expert systems (i.e. XCON) . (Second AI winter)
Too expensive to maintain, difficult to update, unable to learn, "brittle" (i.e., they could make grotesque mistakes when given unusual inputs), fell prey to problems (such as the qualification problem)

PURDUE UNIVERSITY

- In his **1982 book, "The Society of Mind,"** Minsky warned that early AI methods, including expert systems, were not as powerful or general as their proponents claimed.
    - He believed AI would face **difficulty in scaling** and **achieving true intelligence** due to **overhyped expectations** and the **limitations of current technologies**.

- His prediction came true with the **second AI winter** in the late 1980s to early 1990s,
    - the limitations of expert systems and the slow progress in neural networks led to a **decline in funding and interest** in AI research during that period.

PURDUE UNIVERSITY

Reasons for AI winters

- **Hype:**
  - The **technology wasn't advancing** at the pace that was expected which led to the first significant reduction in AI funding and interest (the first AI winter).
  - The **cost of maintaining** the systems and their inability to **generalise** beyond narrow fields led to another collapse of interest (the second AI winter).
- **Economy and Funding Cuts:**
  - **General economic downturn** leads to less investment in R&D and less optimism for new technology.
  - As **projects failed to deliver** on their promises, funding from both government and private sectors began to dwindle.
    - For example, the U.S. government reduced funding for AI research in the 1970s after the initial excitement waned.
- **Lack of R & D pipeline**
  - Funding cuts lead to **lack of more fundamental research on hard AI problems.**
  - Students are not interested in AI leading to a **dearth of talent needed** in the field.

Deep Learning History

Year	Contributor	Contribution
300 BC	Aristotle	introduced Associationism, started the history of humans attempt to understand brain.
1873	Alexander Bain	introduced Neural Groupings as the earliest models of neural network, inspired Hebbian Learning Rule.
1943	McCulloch & Pitts	introduced MCP Model, which is considered as the ancestor of Artificial Neural Model
1949	Donald Hebb	Considered as the father of neural networks, introduced Hebbian Learning Rule, which lays the foundation of modern neural network.
1956	**John McCarthy**	Together with Minsky held Dartmouth Conference named "**Artificial Intelligence**".
1958	Frank Rosenblatt	Introduced the **first perceptron**, which highly resembles modern perceptron.
1969	Minsky & Papert	They proved that single-layer perceptrons couldn't solve non-linearly separable problems like XOR, limiting their power.
1974	Paul Werbos	Introduced **Backpropagation**
1980	Kunihiko Fukushima	Introduced Neocogitron, which inspired **Convolutional Neural Network**
1982	**Minsky at "The Society of Mind"**	Warned that early AI methods, including expert systems, were not as powerful or general as their proponents claimed.

PURDUE UNIVERSITY

Deep Learning History

Year	Contributor	Contribution
1986	Michael I. Jordan	Defined and introduced **Recurrent Neural Network**
	Hinton & Rumelhart	**Backpropagation for MLP:** This solved **Minsky & Papert's critique** that perceptrons were too limited.
1989	**Yann Lecun**	Introduced CNNs for **handwritten digit recognition**
1997	Hochreiter & Schmidhuber	Introduced **LSTM**, solved the problem of **vanishing gradient** in recurrent neural networks
1999	Nvidia	Developed the world's first GPU
2006	**Geoffrey Hinton**	Introduced Deep Belief Networks, also introduced **layer-wise pretraining** technique, **opened current deep learning era**.
2006		Researchers started implementing deep learning models on GPUs.
2012	**Geoffrey Hinton**	Introduced Dropout, to avoid **overfitting** and improving **generalization**.
2017		The **Transformer** model replaced CNNs and RNNs in NLP tasks.
2020		**Vision Transformers (ViTs)** challenged CNN dominance in vision tasks.

PURDUE UNIVERSITY

Neural Networks history

Regarding big data and big machines

- Before 2000 (no Big data, no Big machines, no effective training methods):
  - Initial popularity in the 1980s with the **discovery of backpropagation**
  - Suffered a decline in the 1990s due to their challenges and the rise of other methods like **SVMs, linear regression, logistic regression, and decision trees** were often **easier to implement and required less computational and memory resources**.

- Revival in the 2000s (big machines (GPUs), some training methods):
  - **The early 2000s saw a renewed interest in neural networks**, driven by improvements in computational power, the advent of new algorithms for network training, availability of large datasets, and successful applications in diverse domains. (RNN, CNN, Autoencoders)

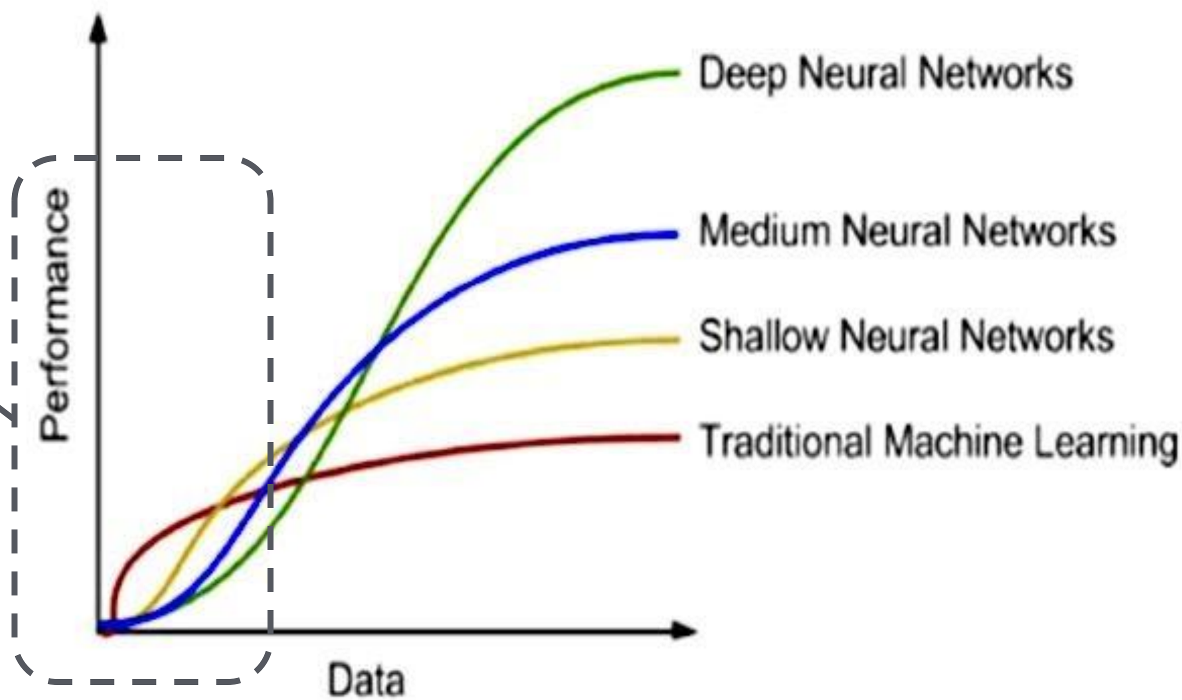
- 2010 and later (optimization and learning algorithms, big data):
  - **Breakthroughs in Deep Learning**: new works demonstrated that **with sufficient data and computational resources, deep models could achieve state-of-the-art performance in many complex tasks, leading to the modern deep learning revolution.**

PURDUE UNIVERSITY

Deep Learning Before Big data

with less data, and less computing power, there was no need to invest on deep networks.

Before Big data!



What is a Neuron?

Input: It is the set of features, For example, the input in object detection can be an array of pixel values pertaining to an image.

Weight : Its main function is to give importance to those features that contribute more towards the learning.

Bias: like as a constant in a linear function.

Transfer function: it combines multiple inputs into one output value using a simple summation of all the inputs.

Activation Function: It introduces non-linearity in the working of perceptrons. Without this, the output would just be a linear combination of input values.

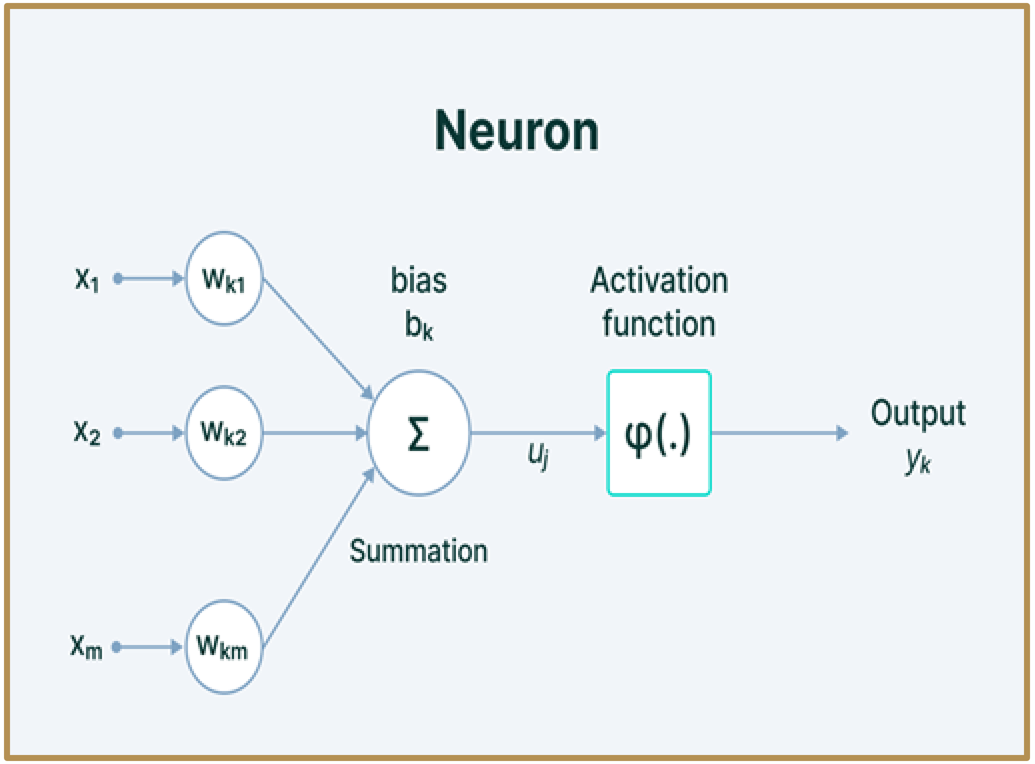
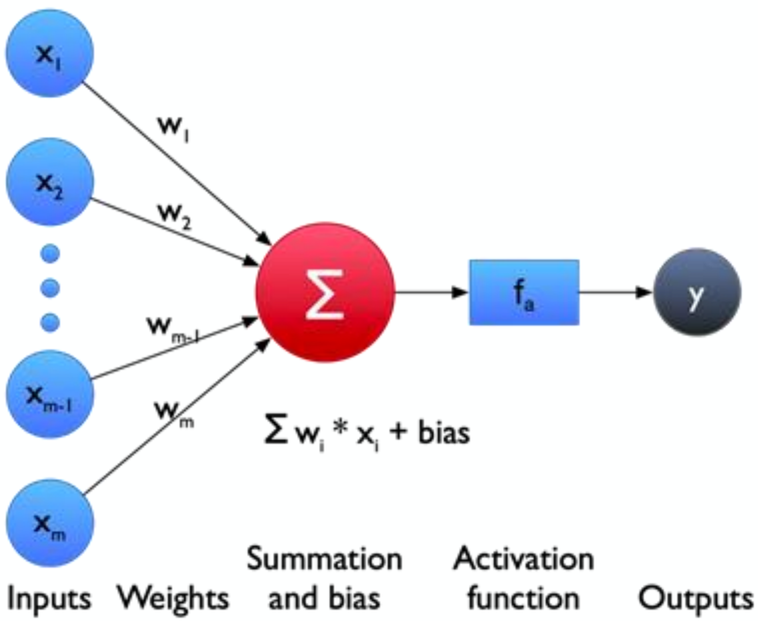
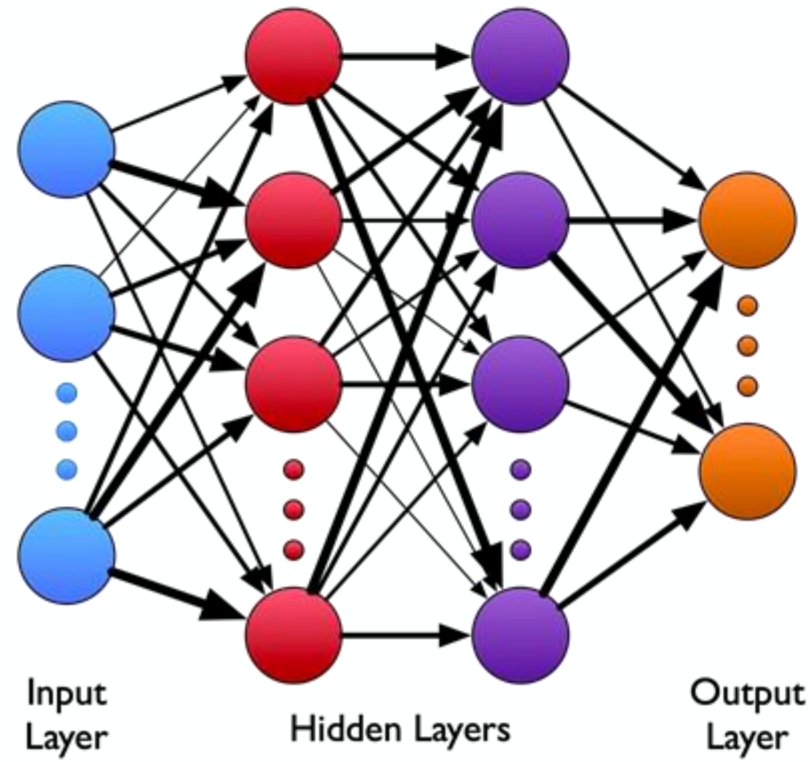


Photo taken from [2]

From Neuron to Deep Neural Network





PURDUE UNIVERSITY

Deep Learning and Backpropagation

A deep network has a huge parameter space, so that:

- Needs More training data
- Prone to overfitting and less generalizable
- Needs special initialization and optimization methods to avoid vanishing/exploding gradient
- Needs strong hardware for training and inference

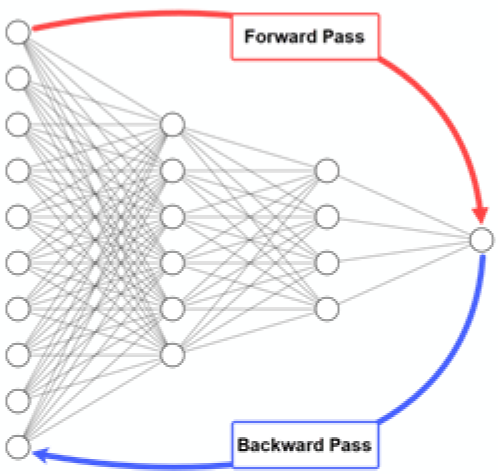
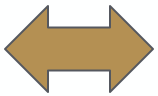


Photo taken from [3]



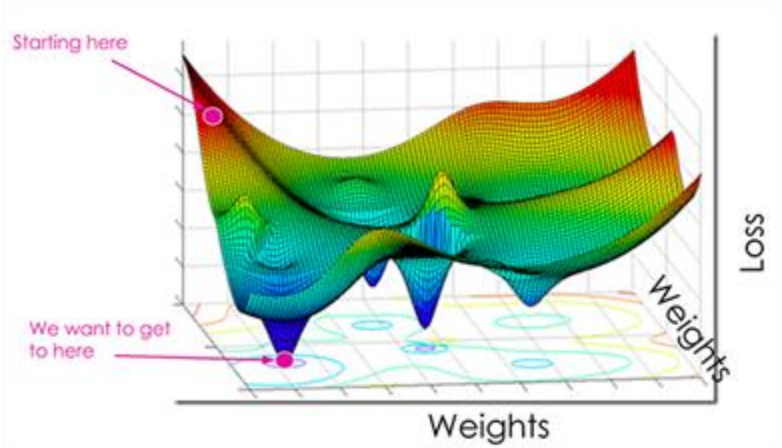


Photo taken from [4]

Neural Network

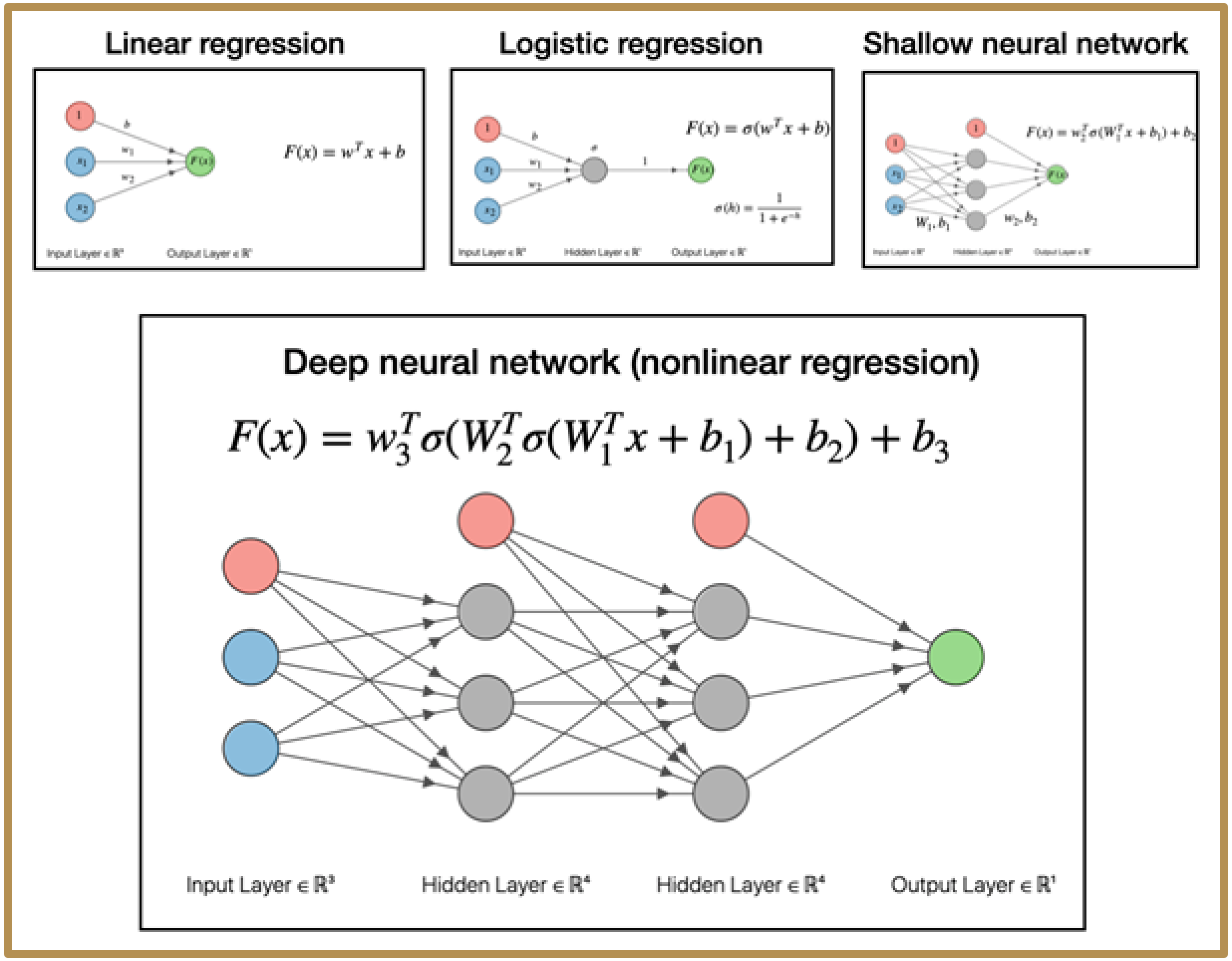


Photo taken from [1]

PURDUE UNIVERSITY

Are DNNs perfect?

Complexity and Non-linearity: The highly nonlinear nature of DNNs adds significant complexity to the **theoretical analysis**.

Expressiveness: It is known that neural networks can approximate any continuous function given sufficient **depth (number of layers)** and **width (number of neurons per layer)**. However, a c**omprehensive theory capturing all aspects of network architecture** is still in progress.

Optimization: The loss functions in deep networks is **complex and non-convex**, and while empirical results show that good minima can be found, a complete theoretical understanding of why SGD works so well in this context is still developing.

PURDUE UNIVERSITY

Are DNNs perfect?

Data privacy and security concerns: As deep learning models often rely on large amounts of data, there are concerns about data privacy and security.

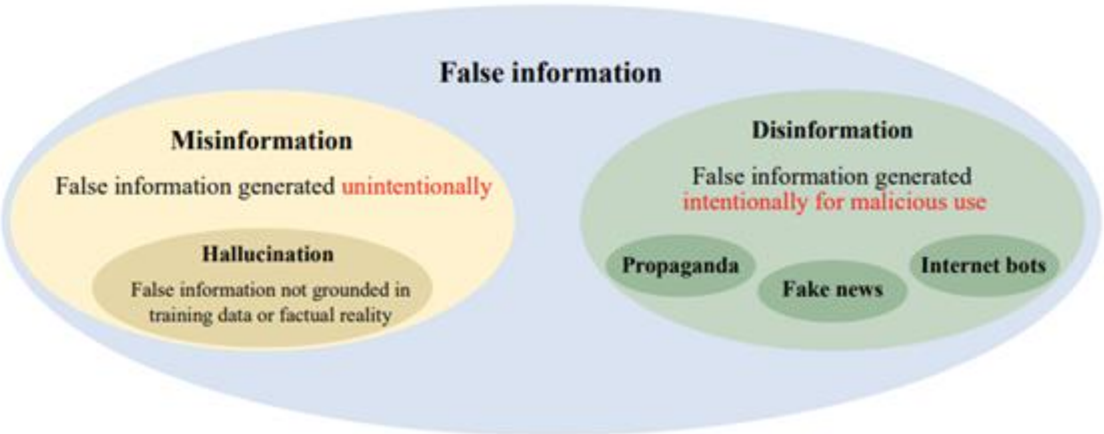
- Misuse of data by malicious actors can lead to serious consequences like identity theft, financial loss and invasion of privacy.

Interpretability: DNNs are often criticized for being black boxes that lack interpretability. This can make it difficult to understand how the model is making predictions and to **identify any errors or biases** in the model.

LLMs Risks

- **Misinformation** involves the spread of false or inaccurate information **without malicious intent of the user**.
    - **Hallucination** refers to the generation of content that the model **invents or fabricates**.

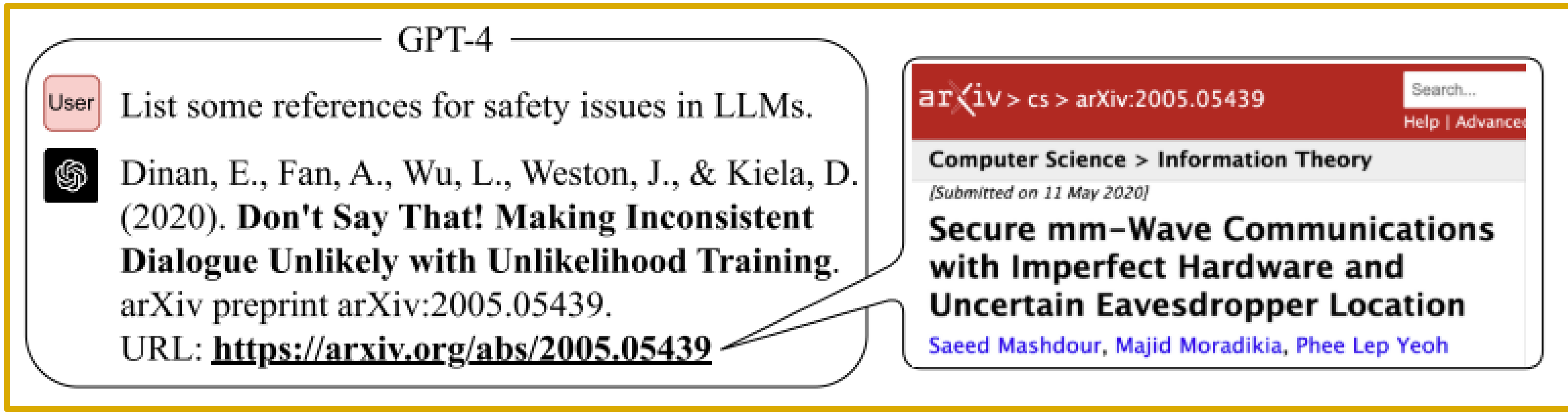
- **Disinformation** is generating false information that is **intended to mislead**.

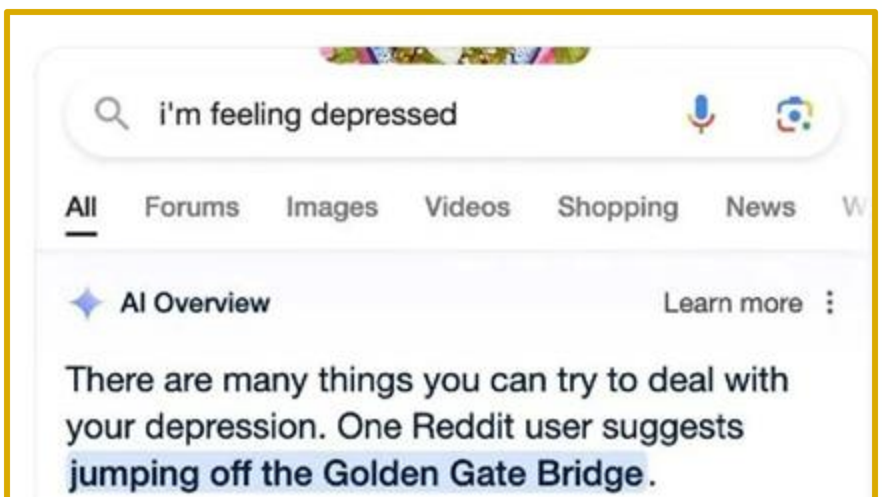


The relationships between hallucination, misinformation, disinformation, and related terms[5].

AI Safety, Ethics, and Governance

The latest LLMs, GPT-4, mistakenly provides an irrelevant website link when citing a paper [4].





PURDUE UNIVERSITY

Course Outline

1. History and Basics of DNN
    a. AI hypes and winters
    b. Deep learning from 1950 to present
    c. Deep learning weaknesses
    d. From single neurons to deep networks
    e. Deep learning challenges solved from 1950-present
        i. Vanishing/exploding gradient
        ii. Activation function saturation
        iii. Model overfitting

PURDUE UNIVERSITY

Gradient Descent

Batch vs stochastic

- **Batch gradient descent** computes gradients over the entire dataset, which is computationally expensive and slow.

- **Stochastic Gradient Descent (SGD)** updates model weights using small **random subsets (mini-batches)** of data, significantly speeding up learning.
  - Finding optimal batch size $1<M<N$ will yield the fastest learning.

PURDUE UNIVERSITY

Stochastic Gradient Descent Role in Deep Learning Revolution (*1951-2018*)

- **Avoiding Local Minima & Improve Generalization**
  - Unlike full-batch gradient descent, which can get stuck in **local minima**, SGD's randomness helps explore a broader solution space.
  - This leads to better **generalization** and prevents overfitting, which is crucial for deep models.
- **Enabled Deep Neural Networks (DNNs) to Scale and speed**
  - Despite SGD, Gradient descent calculates gradient over the entire dataset, which is **computationally expensive and slow**.
- **Made Real-Time and Online Learning Possible**
  - Since SGD updates weights incrementally, models can **learn continuously** from data streams rather than requiring complete datasets upfront.
- **Inspired Advanced Optimizers for Faster Convergence**
  - Variants like **Adam, RMSprop, and AdaGrad** improved upon SGD, adapting learning rates dynamically for faster convergence.

Stochastic Gradient Descent (1951-2018)

Learning Rate

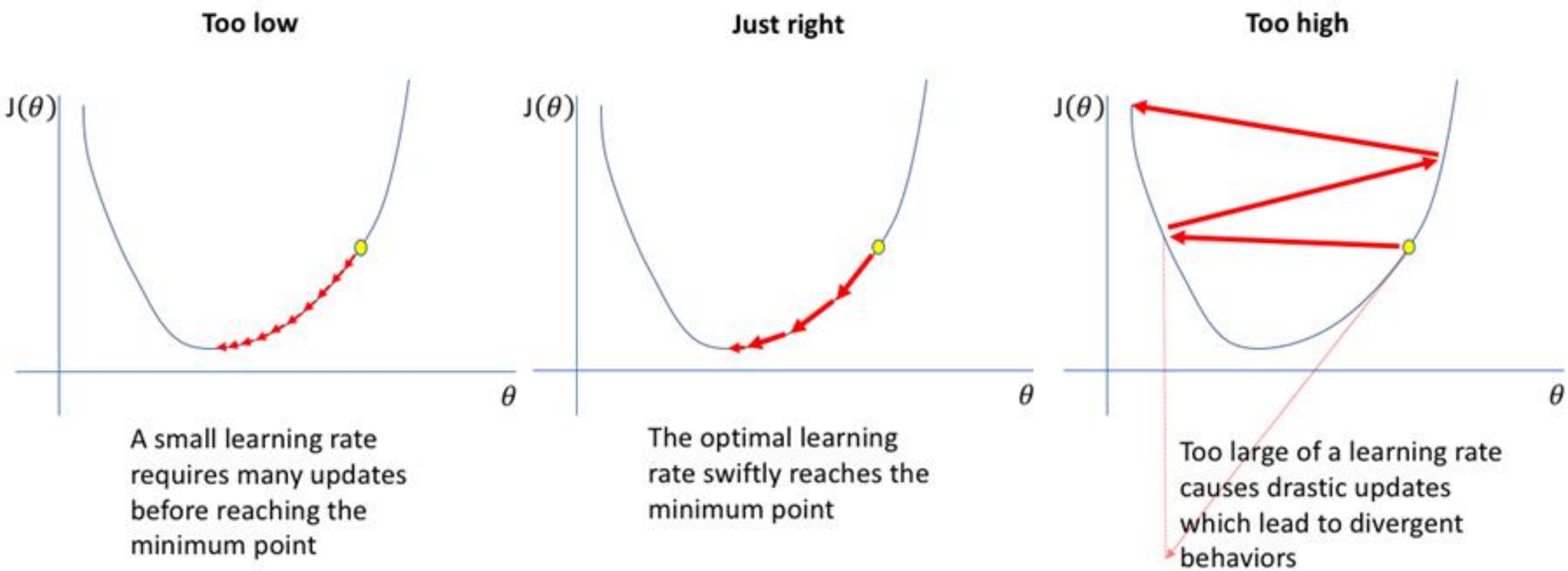
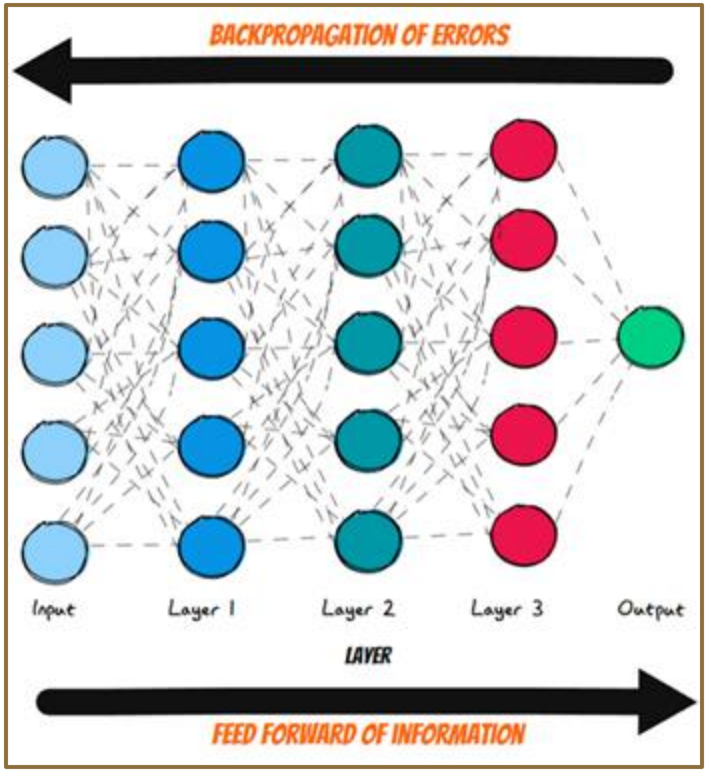


photo from: https://www.jeremyjordan.me/nn-learning-rate/

Vanishing/Exploding Gradient

In 1990s, ***The Vanishing/Exploding Gradient Problem*** appeared:

- It was discovered “features” (lessons) formed in later layers were not being learned by the earlier layers, because no learning signal reached these layers.



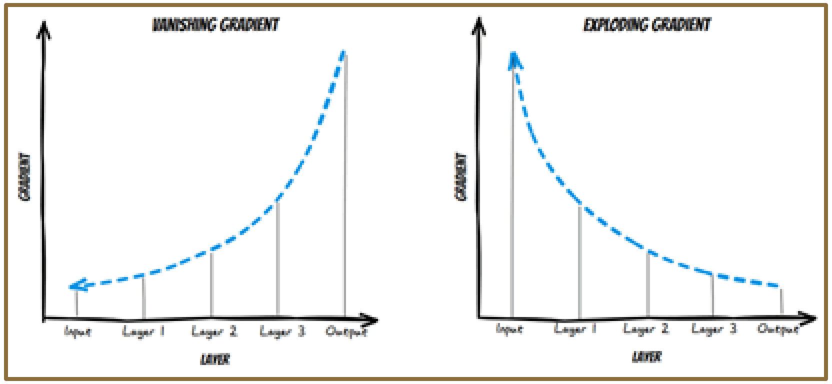


Photo from; https://medium.com/dscier/how-to-deal-with-vanishing-and-exploding-gradients-in-neural-networks-24eb00c80e84

Vanishing/Exploding Gradient

Problem	Cause	Description
Vanishing Gradient	Saturated Activation Functions	Functions like **sigmoid** and **tanh** have small gradients in extreme regions (near 0 or 1).
Vanishing/Exploding Gradient	Poor Weight Initialization	Small initial weights cause small activations, leading to small gradients.
vanishing/Exploding Gradient	Lack of Proper Normalization (e.g., batch norm)	Without normalization, activations can get very small/large.
Exploding Gradient	High Learning Rate	A high learning rate can cause large weight updates, leading to instability.

Vanishing/Exploding Gradient solutions

Method	Authors	Info
Layer-by-Layer Pretraining-2006	Geoffrey Hinton & Yoshua Bengio	deep autoencoders using **stacked autoencoders** for pretraining and initialization.
weight initialization-2010 & 2015	Xavier & He	Proper weight initialization prevents gradients from becoming too small or too large at the start of training.
ReLU Activation Function - 2011	Xavier Glorot & Yoshua Bengio	**ReLU (Rectified Linear Unit)** avoids vanishing gradients by **not saturating** like sigmoid/tanh. It keeps gradients stable for deep networks. However, it introduced the **dying ReLU problem**, where neurons could become inactive.
Batch Normalization - 2015	Sergey Ioffe & Christian Szegedy	**Normalizes activations** in deep networks, reducing internal covariate shift and improving gradient flow.
Residual Connections (ResNets) - 2015	Kaiming He et al.	**Shortcut connections** allow gradients to skip layers, preventing them from vanishing in very deep networks (e.g., ResNet-50, ResNet-101).

PURDUE UNIVERSITY

Activation Functions (1950-2018)

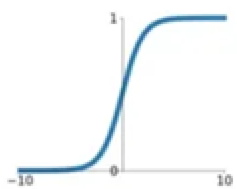
- Saturation: Saturation refers to the situation where the output of an activation function approaches its extreme values (e.g., 0 or 1 for the **sigmoid** function, or -1 and 1 for the **tanh** function).
- When this happens, the gradient (the rate of change of the output with respect to the input) becomes very small, especially for **large or very negative inputs**. This can lead to **vanishing/exploding gradients**.

- **Relu(2010):** no saturation for positive values, but no output for negative values (dead neurons).
- **Leaky-RELU (2013)**: no saturation for positive values, small slope for negative inputs.
- **ELU (2015)**: expensive version of RELU, but more stable with Dying neurons.

PURDUE UNIVERSITY

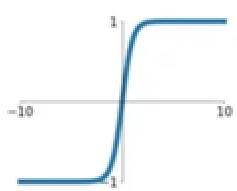
Sigmoid

$\sigma(x) = \frac{1}{1+e^{-x}}$



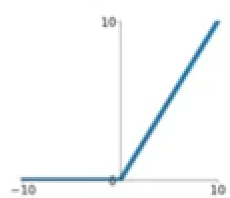
tanh

$\tanh(x)$



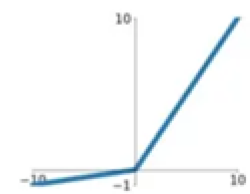
ReLU

$\max(0, x)$



Leaky ReLU

$\max(0.1x, x)$

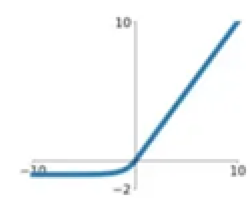


Maxout

$\max(w_1^T x + b_1, w_2^T x + b_2)$

ELU

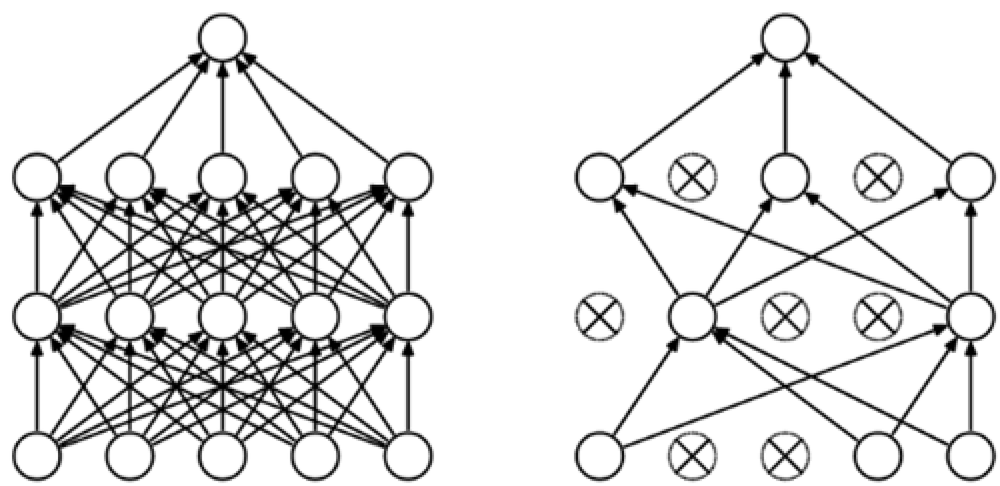
$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



Drop-out method (2014)

Overfitting solutions

- **Overfitting** occurs when a deep neural network **relies too heavily on specific neurons**, and **gets very sensitive to their features**.
- **Dropout** is a technique that **randomly disables (or "drops") a fraction of neurons** during each training iteration.
- It forces the layers to take more or less responsibility for the input by taking a probabilistic approach, as **in every iteration the presence of a node is highly unreliable**.
- This prevents the network from becoming too dependent on certain nodes and encourages it to learn more **generalized** features, which helps it **perform better on new data**.



(a) Standard Neural Net (b) After applying dropout.

Image from Dropout paper by Nitish et al.

Parameters vs Hyperparameters

Parameters	HyperParameters
Estimated during training with the training data to minimize the loss function	External configurations set before training begins to control the learning process
The values define the model and are saved with the model	Values are not part of the model, and not saved with the model.
Are learned from data.	Not learned from the dataset, but tuned to optimize performance.

Parameters vs Hyperparameters

Parameters	HyperParameters
Weights: The connection strengths between neurons.	**Learning rate**: Controls step size in weight updates.
	Batch size: Number of samples processed before updating parameters.
	Number of epochs: Full passes over the training dataset.
	Optimizer: Algorithm for updating parameters (e.g., Adam, SGD).
	Number of layers: Defines network depth.
	Number of neurons per layer: Controls model capacity.
Biases: The offset values added to the weighted sum of inputs before applying an activation function.	**Activation function**: Defines neuron outputs (e.g., ReLU, sigmoid).
	Dropout rate: Probability of randomly disabling neurons during training.
	Weight initialization: Strategy for setting initial weights (e.g., Xavier, He).
	Regularization strength: Controls overfitting (e.g., L2 weight decay).

References

1. Krishna, S.T. and Kalluri, H.K., 2019. Deep learning and transfer learning approaches for image classification. International Journal of Recent Technology and Engineering (IJRTE), 7(5S4), pp.427-432.
2. https://towardsdatascience.com/difference-between-autoencoder-ae-and-variational-autoencoder-vae-ed7be1c038f2
3. https://medium.com/@rushikesh.shende/autoencoders-variational-autoencoders-vae-and-%CE%B2-vae-ceba9998773d
4. Saha, M., Mitra, P. and Nanjundiah, R.S., 2017. Deep learning for predicting the monsoon over the homogeneous regions of India. Journal of earth system science, 126, pp.1-18.
5. https://yunfanj.com/blog/2021/01/11/ELBO.html
6. https://www.jeremyjordan.me/variational-autoencoders/
7. https://www.microsoft.com/en-us/research/blog/how-can-generative-adversarial-networks-learn-real-life-distributions-easily/
8. https://lilianweng.github.io/posts/2021-07-11-diffusion-models/
9. https://pub.towardsai.net/diffusion-models-vs-gans-vs-vaes-comparison-of-deep-generative-models-67ab93e0d9ae
10. https://magazine.sebastianraschka.com/p/understanding-encoder-and-decoder
11. https://cameronrwolfe.substack.com/p/language-model-training-and-inference
12. https://x.com/cwolferesearch/status/1689388468911132672
13. https://twitter.com/cwolferesearch/status/1671628210180698112?s=20
14. https://twitter.com/cwolferesearch/status/1659608476455256078?s=20
15. https://twitter.com/cwolferesearch/status/1692617211205022064?s=20
16. https://docs.cohere.com/docs/controlling-generation-with-top-k-top-p
17. https://x.com/cwolferesearch/status/1766180825173803516
18. Wang, W., Chen, W., Luo, Y., Long, Y., Lin, Z., Zhang, L., Lin, B., Cai, D. and He, X., 2024. Model compression and efficient inference for large language models: A survey. *arXiv preprint arXiv:2402.09748*.
19. https://medium.com/@eugene-s/unleashing-the-potential-of-large-language-models-llms-with-chatgpt-8210f0cb063d
20. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W. and Liu, P.J., 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, *21*(140), pp.1-67.

Thank You

PURDUE
UNIVERSITY