

Artificial Neural Network: Architecture, Learning Algorithms, Applications and Challenges

S.B. Bele¹, Megha D. Angnani², Ishwari P. Telkhade³, Astha V. Thakare⁴

Professor, MCA Department¹

Students, MCA Department²⁻⁴

Vidya Bharati Mahavidyalaya, Amravati, Maharashtra, India

Abstract: Artificial neural networks (ANNs) are computational models inspired by the way biological nervous systems process information. Over the past eight decades they have grown from simple threshold units into deep, billion-parameter systems that underpin modern computer vision, language processing, healthcare analytics, finance and autonomous systems. This paper presents a structured review of ANNs covering their basic building blocks, the major architectures (feedforward, convolutional, recurrent, attention-based, generative and self-organising networks), the principal learning algorithms (supervised, unsupervised, semi-supervised and reinforcement learning together with gradient-based optimisers and regularisation techniques), representative real-world applications, and the open challenges that limit their wider adoption. Comparative tables and schematic diagrams are provided to summarise design choices. The review concludes that, while ANNs deliver state-of-the-art accuracy in many domains, issues of data dependency, interpretability, robustness, energy consumption and ethics remain unresolved and define the most promising directions for future research.

Keywords: artificial neural network, deep learning, backpropagation, convolutional neural network, recurrent neural network, transformer, optimisation, applications.

I. INTRODUCTION

An artificial neural network is a parallel, distributed information-processing system made up of many simple units called neurons that are joined by weighted connections. Knowledge is not programmed explicitly; it is stored in the connection weights and acquired by adjusting them in response to examples. This data-driven property makes ANNs attractive for problems where hand-written rules are impractical, such as recognising faces, translating languages or forecasting demand [19], [24].

The field has passed through several waves of enthusiasm. Early threshold-logic models [1] and the perceptron [2] were followed by a period of limited progress, then by the popularisation of backpropagation [4]. Since 2012, availability of large labelled datasets, graphics processors and improved training tricks has produced a rapid growth in what is now called deep learning [15], [16]. Today ANNs are embedded in consumer products, clinical tools and industrial control systems.

Despite this success, practitioners face persistent questions: which architecture suits which data, how should the network be trained efficiently, and how can its decisions be trusted? The objective of this paper is to answer these questions in a single, consolidated reference. The main contributions are: (i) a concise description of ANN components and architectures; (ii) a comparison of learning algorithms and optimisers; (iii) a survey of application domains; and (iv) a discussion of challenges and future research directions.

The remainder of the paper is organised as follows. Section II reviews related literature, Section III describes the methodology, architectures and learning algorithms, Section IV presents applications, Section V discusses challenges, and Sections VI and VII conclude with future scope.



II. LITERATURE REVIEW

The mathematical foundation of ANNs was laid by McCulloch and Pitts [1], who showed that networks of binary threshold neurons can compute logical functions. Rosenblatt [2] introduced the perceptron together with a learning rule, but single-layer models could solve only linearly separable problems. Hopfield [3] revived interest by demonstrating that recurrent networks can act as associative memories with energy-based dynamics. A decisive step came when Rumelhart, Hinton and Williams [4] showed that the error gradient can be propagated efficiently through hidden layers, enabling multilayer networks to be trained. Theoretical support followed with the universal approximation result of Hornik et al. [5], which states that a feedforward network with a single sufficiently wide hidden layer can approximate any continuous function on a compact domain.

Convolutional networks were applied successfully to handwritten digit recognition by LeCun et al. [6], introducing weight sharing and local receptive fields. Recurrent networks were made practical for long sequences through the gated memory cell of Hochreiter and Schmidhuber [7]. Layer-wise pre-training of deep belief networks [8] and, later, the large-scale ImageNet result of Krizhevsky et al. [9] marked the beginning of the deep-learning era. Residual connections [17] allowed networks with hundreds of layers to be trained, and attention-based Transformers [21] replaced recurrence in most language tasks, leading to large pre-trained models such as BERT [22].

Parallel research addressed training methodology: dropout [11] and batch normalisation [14] improved generalisation and stability, while the Adam optimiser [13] became a common default. Generative adversarial networks [10] opened a new line of work on synthesis. Comprehensive surveys are provided by Schmidhuber [15], LeCun et al. [16] and Goodfellow et al. [19]. More recent studies emphasise the limits of ANNs, including susceptibility to adversarial examples [12] and the need for explanation methods [18]. The key milestones are summarised in Table I.

TABLE I: SUMMARY OF KEY MILESTONES IN ANN RESEARCH

Year	Contribution	Significance
1943	Threshold neuron model [1]	First formal neuron; logic computation
1958	Perceptron [2]	First trainable single-layer network
1982	Hopfield network [3]	Associative memory, energy-based recurrent model
1986	Backpropagation [4]	Efficient training of multilayer networks
1997	LSTM [7]	Learning long-term temporal dependencies
1998	LeNet-5 [6]	Practical convolutional network for digits
2012	Deep CNN on ImageNet [9]	Start of the modern deep-learning era
2014	GAN [10]	Adversarial generative modelling
2016	ResNet [17]	Very deep networks via skip connections
2017	Transformer [21]	Attention-based sequence modelling

III. METHODOLOGY

A. Basic Structure of an Artificial Neuron

An artificial neuron receives inputs $x_1 \dots x_n$, multiplies each by a weight w_i , adds a bias b and passes the sum through a nonlinear activation function f . Its output is $y = f(\sum w_i x_i + b)$. The weights represent the strength of the connections, the bias shifts the decision threshold, and the activation introduces the nonlinearity without which a stack of layers would collapse into a single linear map. Fig. 1 illustrates the model.



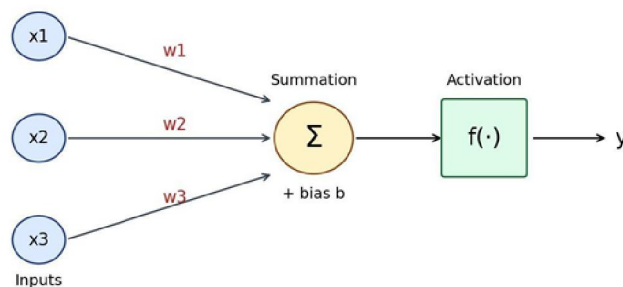


Fig. 1. Model of an artificial neuron

Common activation functions are plotted in Fig. 2. Sigmoid and tanh are smooth but saturate for large inputs, causing small gradients; ReLU is inexpensive and mitigates saturation for positive inputs, and its leaky variant avoids permanently inactive units.

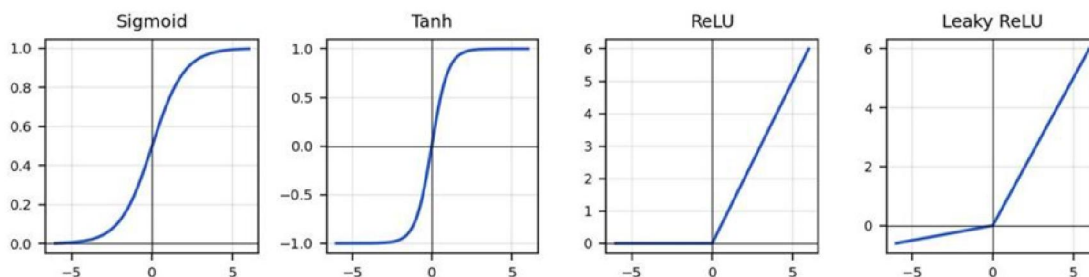


Fig. 2. Commonly used activation functions

B. Network Architecture

Neurons are arranged in layers. A multilayer perceptron (MLP) contains an input layer, one or more hidden layers and an output layer, with every neuron connected to all neurons in the next layer (Fig. 3). Depth (number of layers) and width (neurons per layer) control model capacity. Beyond the MLP, specialised architectures exploit the structure of the data, as compared in Table II.

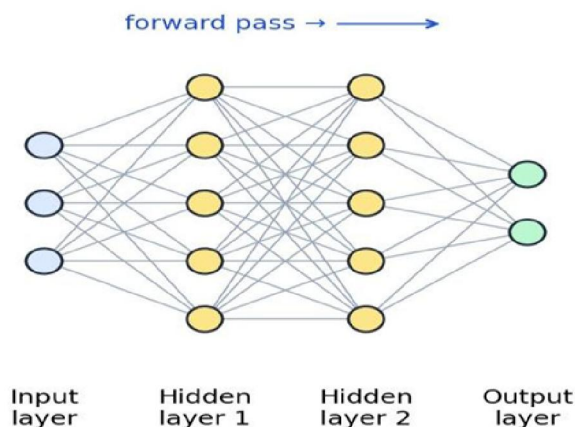


Fig. 3. Architecture of a fully connected multilayer perceptron



TABLE II
Comparison of Major ANN Architectures

Architecture	Key idea	Typical data	Example uses
Feedforward / MLP	Fully connected layers, one-way signal flow	Tabular, fixed-length vectors	Classification, regression
CNN	Local filters with shared weights and pooling	Images, video, spectrograms	Object detection, medical imaging
RNN / LSTM / GRU	Recurrent state with gated memory	Time series, text, speech	Forecasting, speech recognition
Transformer	Self-attention over all positions in parallel	Text, images, multimodal	Translation, chatbots, summarisation
Autoencoder	Encoder-decoder with bottleneck	Any; unlabeled data	Denoising, anomaly detection
GAN	Generator versus discriminator game	Images, audio	Data synthesis, super-resolution
Self-organising map	Competitive, topology-preserving learning	Unlabeled high-dim. data	Clustering, visualisation

C. Learning Algorithms

Learning means finding weights that minimise a loss function L that measures the gap between predictions and targets, for example mean-squared error for regression or cross-entropy for classification. The standard procedure, shown in Fig. 4, is gradient descent with backpropagation. In the forward pass the input is propagated to produce an output; the loss is evaluated; the chain rule is then used to compute $\partial L / \partial w$ for every weight from the output layer backwards; finally the weights are updated as $w \leftarrow w - \eta \partial L / \partial w$, where η is the learning rate [4], [19].

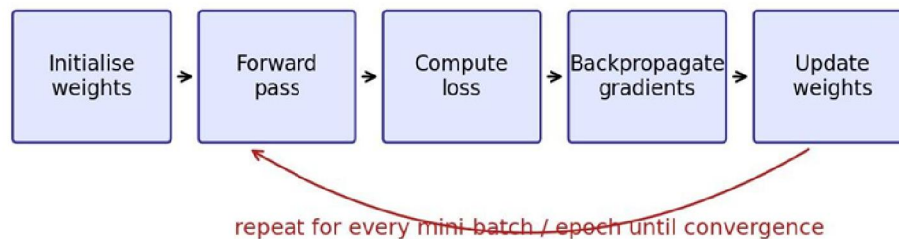


Fig. 4. Training cycle of a neural network using backpropagation

Learning paradigms are categorised by the type of feedback available. In supervised learning every example has a label; in unsupervised learning the network discovers structure by itself; semi- and self-supervised learning combine a small labelled set with a large unlabelled one; and in reinforcement learning an agent learns from scalar rewards obtained through interaction [24]. Table III lists common paradigms and optimisers, and Table IV lists regularisation techniques used to control overfitting.



TABLE III
LEARNING PARADIGMS AND OPTIMISATION ALGORITHMS

Method	Description	Remarks
Supervised learning	Minimise loss on labelled input–output pairs	Most widely used; needs labels
Unsupervised learning	Learn representations or clusters without labels	Autoencoders, SOM, GAN
Reinforcement learning	Maximise cumulative reward via trial and error	Games, robotics [23]
Stochastic gradient descent	Update using mini-batch gradient estimates	Simple; needs tuned η
Momentum / Nesterov	Accumulates past gradients to smooth updates	Faster convergence
RMSprop / Adam [13]	Per-parameter adaptive learning rates	Robust default choice

TABLE IV
REGULARISATION AND TRAINING-STABILISATION TECHNIQUES

Technique	Principle	Benefit
L1 / L2 penalty	Adds weight-magnitude term to the loss	Simpler, smoother models
Dropout [11]	Randomly disables neurons during training	Reduces co-adaptation
Batch normalisation [14]	Normalises layer inputs per mini-batch	Faster, stable training
Early stopping	Stops when validation loss stops improving	Prevents overfitting
Data augmentation	Creates modified copies of training data	Better generalisation

D. Design and Evaluation Workflow

A practical ANN project follows a consistent workflow: (1) collect and clean data; (2) split it into training, validation and test sets; (3) normalise features; (4) choose an architecture and loss function; (5) train with a suitable optimiser while monitoring validation performance; (6) tune hyper-parameters such as learning rate, batch size and depth; and (7) report final performance on the untouched test set using metrics appropriate for the task, such as accuracy, precision, recall, F1-score, ROC-AUC or root-mean-square error. Reproducibility is improved by fixing random seeds and documenting software versions.

IV. APPLICATIONS

ANNs are used wherever large amounts of data contain patterns that are hard to describe with explicit rules. Table V summarises major domains.

TABLE V
MAJOR APPLICATION AREAS OF ANNS

Domain	Typical tasks	Common architecture
Computer vision	Image classification, detection, face recognition, segmentation	CNN, Vision Transformer
Natural language processing	Translation, question answering, sentiment analysis, chatbots	Transformer, LSTM
Healthcare	Disease detection from scans, e.g. skin lesions [20]; drug discovery	CNN, autoencoder



Domain	Typical tasks	Common architecture
Finance	Credit scoring, fraud detection, stock-trend prediction	MLP, LSTM
Speech and audio	Speech recognition, speaker identification, music generation	RNN, CNN, Transformer
Robotics and control	Navigation, manipulation, game playing [23]	Deep RL, CNN
Engineering and energy	Load forecasting, fault diagnosis, predictive maintenance	MLP, RNN
Agriculture and environment	Crop disease detection, weather and pollution forecasting	CNN, LSTM

In vision, deep CNNs now match or exceed human accuracy on several benchmark tasks [9], [17]. In medicine, a CNN classified skin lesions with accuracy comparable to dermatologists [20]. Deep reinforcement learning combined with neural networks defeated a professional Go player [23]. In language, pre-trained Transformer models achieve leading results on translation and comprehension tasks [21], [22]. In finance and energy, recurrent and feedforward models support forecasting and anomaly detection, where they often outperform classical statistical baselines when sufficient data are available.

V. CHALLENGES

Despite impressive performance, several challenges restrict the deployment of ANNs, especially in safety-critical settings. They are summarised in Table VI.

TABLE VI
KEY CHALLENGES AND POSSIBLE MITIGATION STRATEGIES

Challenge	Description	Mitigation
Data dependency	Large labelled datasets are costly; biased data yields biased models	Transfer learning, augmentation, self-supervision
Overfitting	Memorising training data reduces generalisation	Regularisation, more data, cross-validation
Interpretability	Decisions are hard to explain (“black box”)	Explainable AI such as LIME [18]
Adversarial vulnerability	Tiny perturbations can flip predictions [12]	Adversarial training, robust design
Computational cost	Training needs GPUs and high energy	Pruning, quantisation, efficient designs
Vanishing/exploding gradients	Signals shrink or grow through deep layers	ReLU, residual links [17], normalisation
Privacy and ethics	Leakage of personal data, fairness, accountability	Federated learning, audits, regulation

Hyper-parameter selection also remains largely empirical, and models trained in one environment often degrade when the data distribution changes. Addressing these issues is necessary for trustworthy adoption.



VI. CONCLUSION

This paper reviewed artificial neural networks from the single neuron to modern deep architectures, and described the learning algorithms, regularisation techniques, applications and challenges associated with them. The evidence shows that architectural innovations such as convolution, gating, residual connections and attention, together with better optimisation and abundant data, are responsible for the large gains observed in recent years. At the same time, dependence on data, limited transparency, sensitivity to adversarial inputs and high resource consumption show that ANNs are powerful tools but not universal solutions. Careful choice of architecture, disciplined training and rigorous evaluation remain essential for reliable results.

VII. FUTURE SCOPE

Several research directions are likely to shape the next generation of neural networks: (i) explainable and trustworthy AI, giving human-understandable reasons for predictions; (ii) data-efficient learning through self-supervised, few-shot and transfer approaches; (iii) energy-efficient models and neuromorphic hardware that mimic spiking behaviour of biological neurons; (iv) privacy-preserving training such as federated learning on edge devices; (v) robust networks that resist adversarial attacks and distribution shifts; (vi) neuro-symbolic systems that combine learning with logical reasoning; and (vii) multimodal models that jointly process text, images, audio and sensor data. Progress in these areas will extend ANNs to more sensitive domains such as healthcare, transportation and public infrastructure.

REFERENCES

- [1] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bulletin of Mathematical Biophysics*, vol. 5, no. 4, pp. 115–133, 1943.
- [2] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychological Review*, vol. 65, no. 6, pp. 386–408, 1958.
- [3] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proc. Natl. Acad. Sci. USA*, vol. 79, no. 8, pp. 2554–2558, 1982.
- [4] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, 1986.
- [5] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989.
- [6] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [7] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [8] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.
- [9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems (NIPS)*, vol. 25, 2012.
- [10] I. Goodfellow *et al.*, "Generative adversarial nets," *Advances in Neural Information Processing Systems*, vol. 27, 2014.
- [11] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, pp. 1929–1958, 2014.
- [12] C. Szegedy *et al.*, "Intriguing properties of neural networks," *Int. Conf. on Learning Representations (ICLR)*, 2014.
- [13] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *Int. Conf. on Learning Representations (ICLR)*, 2015.



- [14] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *Proc. ICML*, pp. 448–456, 2015.
- [15] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.
- [16] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, 2015.
- [17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proc. IEEE CVPR*, pp. 770–778, 2016.
- [18] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should I trust you? Explaining the predictions of any classifier," *Proc. ACM SIGKDD*, pp. 1135–1144, 2016.
- [19] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
- [20] A. Esteva *et al.*, "Dermatologist-level classification of skin cancer with deep neural networks," *Nature*, vol. 542, pp. 115–118, 2017.
- [21] A. Vaswani *et al.*, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.
- [22] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," *Proc. NAACL-HLT*, pp. 4171–4186, 2019.
- [23] D. Silver *et al.*, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, pp. 484–489, 2016.
- [24] S. Haykin, *Neural Networks and Learning Machines*, 3rd ed. Upper Saddle River, NJ, USA: Pearson, 2009.

