

Title: Neural field

URL: https://en.wikipedia.org/wiki/Neural_field

PageID: 80384572

Categories: Category:Artificial neural networks, Category:Deep learning, Category:Neural network architectures, Category:Neural networks

Source: Wikipedia (CC BY-SA 4.0).

Supervised learning

Unsupervised learning

Semi-supervised learning

Self-supervised learning

Reinforcement learning

Meta-learning

Online learning

Batch learning

Curriculum learning

Rule-based learning

Neuro-symbolic AI

Neuromorphic engineering

Quantum machine learning

Classification

Generative modeling

Regression

Clustering

Dimensionality reduction

Density estimation

Anomaly detection

Data cleaning

AutoML

Association rules

Semantic analysis

Structured prediction

Feature engineering

Feature learning

Learning to rank

Grammar induction

Ontology learning

Multimodal learning

Apprenticeship learning

Decision trees

Ensembles Bagging Boosting Random forest

Bagging

Boosting

Random forest

k -NN

Linear regression

Naive Bayes

Artificial neural networks

Logistic regression

Perceptron

Relevance vector machine (RVM)

Support vector machine (SVM)

BIRCH

CURE

Hierarchical

k -means

Fuzzy

Expectation–maximization (EM)

DBSCAN

OPTICS

Mean shift

Factor analysis

CCA

ICA

LDA

NMF

PCA

PGD

t-SNE

SDL

Graphical models Bayes net Conditional random field Hidden Markov

Bayes net

Conditional random field

Hidden Markov

RANSAC

k -NN

Local outlier factor
Isolation forest
Autoencoder
Deep learning
Feedforward neural network
Recurrent neural network LSTM GRU ESN reservoir computing
LSTM
GRU
ESN
reservoir computing
Boltzmann machine Restricted
Restricted
GAN
Diffusion model
SOM
Convolutional neural network U-Net LeNet AlexNet DeepDream
U-Net
LeNet
AlexNet
DeepDream
Neural field Neural radiance field Physics-informed neural networks
Neural radiance field
Physics-informed neural networks
Transformer Vision
Vision
Mamba
Spiking neural network
Memtransistor
Electrochemical RAM (ECRAM)
Q-learning
Policy gradient
SARSA
Temporal difference (TD)
Multi-agent Self-play
Self-play
Active learning
Crowdsourcing
Human-in-the-loop

Mechanistic interpretability

RLHF

Coefficient of determination

Confusion matrix

Learning curve

ROC curve

Kernel machines

Bias–variance tradeoff

Computational learning theory

Empirical risk minimization

Occam learning

PAC learning

Statistical learning

VC theory

Topological deep learning

AAAI

ECML PKDD

NeurIPS

ICML

ICLR

IJCAI

ML

JMLR

Glossary of artificial intelligence

List of datasets for machine-learning research List of datasets in computer vision and image processing

List of datasets in computer vision and image processing

Outline of machine learning

v

t

e

In machine learning , a neural field (also known as implicit neural representation , neural implicit , or coordinate-based neural network), is a mathematical field that is fully or partially parametrized by a neural network . Initially developed to tackle visual computing tasks, such as rendering or reconstruction (e.g., neural radiance fields), neural fields emerged as a promising strategy to deal with a wider range of problems, including surrogate modelling of partial differential equations , such as in physics-informed neural networks . [1]

Differently from traditional machine learning algorithms, such as feed-forward neural networks , convolutional neural networks , or transformers , neural fields do not work with discrete data (e.g. sequences , images , tokens), but map continuous inputs (e.g., spatial coordinates , time) to continuous outputs (i.e., scalars , vectors , etc.). This makes neural fields not only discretization

independent, but also easily differentiable . Moreover, dealing with continuous data allows for a significant reduction in space complexity , which translates to a much more lightweight network. [1]

Formulation and training

According to the universal approximation theorem , provided adequate learning, sufficient number of hidden units , and the presence of a deterministic relationship between the input and the output, a neural network can approximate any function to any degree of accuracy . [2]

Hence, in mathematical terms, given a field $y = \Phi(x)$, with $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^m$, a neural field Ψ_θ , with parameters θ , is such that: $\Psi_\theta(x) = y \approx \hat{y}$

Training

For supervised tasks, given N examples in the training dataset (i.e., $(x_i, y_i) \in \mathcal{D}_{train}, i = 1, \dots, N$), the neural field parameters can be learned by minimizing a loss function L (e.g., mean squared error). The parameters $\tilde{\theta}$ that satisfy the optimization problem are found as: $\tilde{\theta} = \underset{\theta}{\operatorname{argmin}} \frac{1}{N} \sum_{(x_i, y_i) \in \mathcal{D}_{train}} L(\Psi_\theta(x_i), y_i)$. Notably, it is not necessary to know the analytical expression of Φ , for the previously reported training procedure only requires input-output pairs. Indeed, a neural field is able to offer a continuous and differentiable surrogate of the true field, even from purely experimental data . [1]

Moreover, neural fields can be used in unsupervised settings, with training objectives that depend on the specific task. For example, physics-informed neural networks may be trained on just the residual . [4]

Spectral bias

As for any artificial neural network, neural fields may be characterized by a spectral bias (i.e., the tendency to preferably learn the low frequency content of a field), possibly leading to a poor representation of the ground truth . [5] In order to overcome this limitation, several strategies have been developed. For example, SIREN uses sinusoidal activations , [6] while the Fourier-features approach embeds the input through sines and cosines . [7]

Conditional neural fields

In many real-world cases, however, learning a single field is not enough. For example, when reconstructing 3D vehicle shapes from Lidar data, it is desirable to have a machine learning model that can work with arbitrary shapes (e.g., a car , a bicycle , a truck , etc.). The solution is to include additional parameters, the latent variables (or latent code) $z \in \mathbb{R}^d$, to vary the field and adapt it to diverse tasks. [1]

Latent code production

When dealing with conditional neural fields, the first design choice is represented by the way in which the latent code is produced. Specifically, two main strategies can be identified: [1]

Encoder: the latent code is the output of a second neural network, acting as an encoder. During training, the loss function is the objective used to learn the parameters of both the neural field and the encoder. [8]

Auto-decoding: each training example has its own latent code, jointly trained with the neural field parameters. When the model has to process new examples (i.e., not originally present in the training dataset), a small optimization problem is solved, keeping the network parameters fixed and only learning the new latent variables. [9]

Since the latter strategy requires additional optimization steps at inference time, it sacrifices speed, but keeps the overall model smaller. Moreover, despite being simpler to implement, an encoder may harm the generalization capabilities of the model. [1] For example, when dealing with a physical scalar field $f: \mathbb{R}^2 \rightarrow \mathbb{R}$ (e.g., the pressure of a 2D fluid), an auto-decoder-based conditional neural field can map a single point to the corresponding value of the field, following a learned latent code z . [10] However, if the latent variables were produced by an encoder, it would require access to the entire set of points and corresponding values (e.g. as a regular grid or a mesh graph), leading to a less robust model. [1]

Global and local conditioning

In a neural field with global conditioning, the latent code does not depend on the input and, hence, it offers a global representation (e.g., the overall shape of a vehicle). However, depending on the task, it may be more useful to divide the domain of x in several subdomains, and learn different latent codes for each of them (e.g., splitting a large and complex scene in sub-scenes for a more efficient rendering). This is called local conditioning. [1]

Conditioning strategies

There are several strategies to include the conditioning information in the neural field. In the general mathematical framework, conditioning the neural field with the latent variables is equivalent to mapping them to a subset θ^* of the neural field parameters: $\theta^* = \Gamma(z)$. In practice, notable strategies are:

Concatenation: the neural field receives, as input, the concatenation of the original input x with the latent codes z . For feed-forward neural networks, this is equivalent to setting θ^* as the bias of the first layer and $\Gamma(z)$ as an affine transformation. [1]

Hypernetworks: a hypernetwork is a neural network that outputs the parameters of another neural network. [11] Specifically, it consists of approximating $\Gamma(z)$ with a neural network $\hat{\Gamma}_\gamma(z)$, where γ are the trainable parameters of the hypernetwork. This approach is the most general, as it allows to learn the optimal mapping from latent codes to neural field parameters. However, hypernetworks are associated to larger computational and memory complexity, due to the large number of trainable parameters. Hence, leaner approaches have been developed. For example, in the Feature-wise Linear Modulation (FiLM), the hypernetwork only produces scale and bias coefficients for the neural field layers. [1] [12]

Meta-learning

Instead of relying on the latent code to adapt the neural field to a specific task, it is also possible to exploit gradient-based meta-learning. In this case, the neural field is seen as the specialization of an underlying meta-neural-field, whose parameters are modified to fit the specific task, through a few steps of gradient descent. [13] [14] An extension of this meta-learning framework is the CAVIA algorithm, that splits the trainable parameters in context-specific and shared groups, improving parallelization and interpretability, while reducing meta-overfitting. This strategy is similar to the auto-decoding conditional neural field, but the training procedure is substantially different. [15]

Applications

Thanks to the possibility of efficiently modelling diverse mathematical fields with neural networks, neural fields have been applied to a wide range of problems:

3D scene reconstruction: neural fields can be used to model the properties of 3D scenes (i.e., geometry, appearance, materials, and lighting), in both static and dynamic cases. [1] For example, a neural field can learn signed distance functions (SDFs) [9] or occupancy functions, [16] which provide an efficient and continuous representation of the geometry. Another example is

represented by neural radiance fields (NeRFs), that learn to render 3D scenes, by mapping coordinates and viewing angles to the corresponding radiance and density. [17]

Digital humans : neural fields can be used to model human shape and appearance and can include information on the complex movements of a human body . [1]

Generative modelling : by leveraging conditioning, neural fields can also work as deep generative models. [1]

Image processing : with respect to convolutional neural networks, neural fields offer a continuous representation of the image and, hence, are not limited to the original pixel discretization. [1]

Robotics : the strengths of neural fields in scene reconstruction are also useful in robotics, as navigation requires reconstructing the surroundings from sensor data. Moreover, neural fields can be used for planning and control . [1]

Lossy data compression [1]

Signal processing [1]

Scientific computing : scientific machine learning (SciML) recently emerged as the combination of physics-based and data-driven models, to numerically solve differential equations . [4] In this context, the ability of neural fields to model input and solution in a continuous and differentiable manner is invaluable. For example, physics-informed neural networks (PINNs) use neural fields to include, in the training objective, the residual computed via automatic differentiation . [18] Instead, encode-process-decode architectures (e.g. CORAL), built on conditional neural fields, have been explored as an alternative operator-learning technique. [19] [10]

See also

Artificial intelligence

Machine learning

Neural network (machine learning)

Neural radiance field

Neural operators

References

External links

Brown University's database of neural-field architectures

Neural Radiance Fields: visual computing applications

GitHub-Awesome list of Implicit Neural Representations

v

t

e

History timeline

timeline

Companies

Projects

Parameter Hyperparameter

Hyperparameter

Loss functions

Regression Bias–variance tradeoff Double descent Overfitting

Bias–variance tradeoff
Double descent
Overfitting
Clustering
Gradient descent SGD Quasi-Newton method Conjugate gradient method
SGD
Quasi-Newton method
Conjugate gradient method
Backpropagation
Attention
Convolution
Normalization Batchnorm
Batchnorm
Activation Softmax Sigmoid Rectifier
Softmax
Sigmoid
Rectifier
Gating
Weight initialization
Regularization
Datasets Augmentation
Augmentation
Prompt engineering
Reinforcement learning Q-learning SARSA Imitation Policy gradient
Q-learning
SARSA
Imitation
Policy gradient
Diffusion
Latent diffusion model
Autoregression
Adversary
RAG
Uncanny valley
RLHF
Self-supervised learning
Reflection
Recursive self-improvement

Hallucination
Word embedding
Vibe coding
Machine learning In-context learning
In-context learning
Artificial neural network Deep learning
Deep learning
Language model Large language model NMT
Large language model
NMT
Reasoning language model
Model Context Protocol
Intelligent agent
Artificial human companion
Humanity's Last Exam
Artificial general intelligence (AGI)
AlexNet
WaveNet
Human image synthesis
HWR
OCR
Computer vision
Speech synthesis 15.ai ElevenLabs
15.ai
ElevenLabs
Speech recognition Whisper
Whisper
Facial recognition
AlphaFold
Text-to-image models Aurora DALL-E Firefly Flux Ideogram Imagen Midjourney Recraft Stable Diffusion
Aurora
DALL-E
Firefly
Flux
Ideogram
Imagen
Midjourney

Recraft

Stable Diffusion

Text-to-video models Dream Machine Runway Gen Hailuo AI Kling Sora Veo

Dream Machine

Runway Gen

Hailuo AI

Kling

Sora

Veo

Music generation Riffusion Suno AI Udio

Riffusion

Suno AI

Udio

Word2vec

Seq2seq

GloVe

BERT

T5

Llama

Chinchilla AI

PaLM

GPT 1 2 3 J ChatGPT 4 4o o1 o3 4.5 4.1 o4-mini 5

1

2

3

J

ChatGPT

4

4o

o1

o3

4.5

4.1

o4-mini

5

Claude

Gemini Gemini (language model) Gemma

Gemini (language model)

Gemma
Grok
LaMDA
BLOOM
DBRX
Project Debater
IBM Watson
IBM Watsonx
Granite
PanGu- Σ
DeepSeek
Qwen
AlphaGo
AlphaZero
OpenAI Five
Self-driving car
MuZero
Action selection AutoGPT
AutoGPT
Robot control
Alan Turing
Warren Sturgis McCulloch
Walter Pitts
John von Neumann
Claude Shannon
Shun'ichi Amari
Kunihiko Fukushima
Takeo Kanade
Marvin Minsky
John McCarthy
Nathaniel Rochester
Allen Newell
Cliff Shaw
Herbert A. Simon
Oliver Selfridge
Frank Rosenblatt
Bernard Widrow
Joseph Weizenbaum

Seymour Papert
Seppo Linnainmaa
Paul Werbos
Geoffrey Hinton
John Hopfield
Jürgen Schmidhuber
Yann LeCun
Yoshua Bengio
Lotfi A. Zadeh
Stephen Grossberg
Alex Graves
James Goodnight
Andrew Ng
Fei-Fei Li
Alex Krizhevsky
Ilya Sutskever
Oriol Vinyals
Quoc V. Le
Ian Goodfellow
Demis Hassabis
David Silver
Andrej Karpathy
Ashish Vaswani
Noam Shazeer
Aidan Gomez
John Schulman
Mustafa Suleyman
Jan Leike
Daniel Kokotajlo
François Chollet
Neural Turing machine
Differentiable neural computer
Transformer Vision transformer (ViT)
Vision transformer (ViT)
Recurrent neural network (RNN)
Long short-term memory (LSTM)
Gated recurrent unit (GRU)
Echo state network

Multilayer perceptron (MLP)

Convolutional neural network (CNN)

Residual neural network (RNN)

Highway network

Mamba

Autoencoder

Variational autoencoder (VAE)

Generative adversarial network (GAN)

Graph neural network (GNN)

Category