EUROPEAN PATENT OFFICE
U.S. PATENT AND TRADEMARK OFFICE

CPC NOTICE OF CHANGES 1493

DATE: AUGUST 1, 2023

PROJECT DP11849

The following classification changes will be effected by this Notice of Changes:

Action	Subclass	Group(s)
DEFINITIONS:		
Definitions New:	G06N	3/0455, 3/0464, 3/0475, 3/0499, 3/0895, 3/09, 3/091, 3/092, 3/094, 3/096, 3/098, 3/0985
	G06N	5/025
Definitions Modified:	G06N	3/004, 3/006, 3/008, 3/042, 3/043, 3/044, 3/045, 3/047, 3/048, 3/049, 3/065, 3/08, 3/082, 3/084, 3/086, 3/088, 3/123, 3/126
	G06N	5/01, 5/02, 5/043, 5/045, 5/046, 5/048
	G06N	7/01
	G06N	20/00

No other subclasses/groups are impacted by this Notice of Changes.

This Notice of Changes includes the following *[Check the ones included]:*

1. CLASSIFICATION SCHEME CHANGES
   - ☐ A. New, Modified or Deleted Group(s)
   - ☐ B. New, Modified or Deleted Warning(s)
   - ☐ C. New, Modified or Deleted Note(s)
   - ☐ D. New, Modified or Deleted Guidance Heading(s)
2. DEFINITIONS
   - ☒ A. New or Modified Definitions (Full definition template)
   - ☐ B. Modified or Deleted Definitions (Definitions Quick Fix)
3. ☐ REVISION CONCORDANCE LIST (RCL)
4. ☐ CHANGES TO THE CPC-TO-IPC CONCORDANCE LIST (CICL)
5. ☐ CHANGES TO THE CROSS-REFERENCE LIST (CRL)

2. A. DEFINITIONS (new)

Insert: The following new Definitions.

G06N 3/0455

Definition statement

This place covers:

Architectures learning representations for datasets and comprising two main components: an encoder transforming inputs into a latent space and a decoder transforming this intermediate representation to the outputs. Additional components further process data or embeddings in the input, latent or output spaces. Auto-encoders learn to reconstruct the original representation.

Typical examples of such neural network models: stochastic autoencoders (SAE), denoising autoencoders (DAE), contractive autoencoders (CAE), variational autoencoders (VAE), ladder networks, convolution-deconvolution networks.

G06N 3/0464

Definition statement

This place covers:

Convolutional neural networks, or convolution neural networks, are a specialised type of artificial neural networks that use a mathematical operation called convolution in place of general matrix multiplication in at least one of their layers.

Typical examples or features of such neural network models: pooling layers, convolution layers, feature maps, dilated convolutions, residual networks (ResNet), dense networks (DenseNet), U-Net.

G06N 3/0475

Definition statement

This place covers:

Neural networks having as special feature that they generate candidates from an existing distribution of samples, e.g. generate virtual image, text, code or sound data examples.

Typical examples of such neural network models: generative adversarial networks (GAN), Boltzmann machines, Helmholtz machines, energy-based models, spin-based models, networks based on Ising or Pott models.

G06N 3/0499

Definition statement

This place covers:

Neural networks having as special feature that connections between the nodes do not form a cycle. This network model is structured with only forward connections through which data is transmitted in one direction.

Relationships with other classification places

This group covers only neural networks with feedforward structures that are not provided for by other groups under G06N 3/04.

G06N 3/0895

Definition statement

This place covers:

Learning or training with partial or artificial supervisory signals, e.g. using a training set with a limited number of ground truth labelled data, or by generating pseudo-labels from an unlabelled dataset.

Typical examples of such learning or training methods: barely-supervised learning, co-training, pseudo-labelling, data augmentation, learning with noisy labels, consistency regularisations, FixMatch, MixMatch.

G06N 3/09

Definition statement

This place covers:

Using a labelled data set to train or learn neural network models. These datasets are designed to “supervise” the training or learning of models into classifying, regressing or generally predicting data or outcomes accurately.

Typical examples of such learning or training methods: empirical risk minimisation (ERM), structural risk minimisation (SRM), MixUp, instance-based learning, neural network classifiers, neural network regressors, learning vector quantisation, training-validation-test frameworks.

G06N 3/091

Definition statement

This place covers:

Learning or training relying on human interactions in general (human-in-the-loop), such as active learning techniques querying a user/oracle/teacher to label selected data. The various strategies for interacting/querying humans often aim at minimising the cost of manual labelling, or maximising the accuracy of the predictions.

G06N 3/092

Definition statement

This place covers:

Techniques that enable an agent to learn a policy in an interactive environment by trial and error using feedback from its actions and experiences. The policy optimises a reward/value/utility function, or other reinforcement signals. Reinforcement learning is often modelled as a Markov decision process (MDP). Neural networks may, e.g., be used to represent the policy, or approximate reinforcement signals.

Typical examples or features of such learning or training methods: policy gradient, policy optimisation, policy search, reinforcement learning agents, multi-agent systems, actor-critic, advantage functions, reward functions, utility functions, value functions, Q-values, deep Q-networks (DQN), Q-learning, imitation learning, temporal difference (TD) learning, multi-armed bandit (MAB), A3C algorithms, DDPG algorithms, Dyna algorithms, PPO algorithms, SARSA algorithms.

G06N 3/094

Definition statement

This place covers:

Learning method which aims to trick machine learning models by providing deceptive/distorted/noisy inputs. This includes both the generation and detection of adversarial examples, which are inputs specially created to deceive classifiers, as well as techniques for improving the neural network's robustness to adversarial/poisoning attacks.

Typical examples or features of such learning or training methods: Wasserstein losses, earth mover's distances, adversarial regularisations, fast gradient sign methods (FGSM), projected gradient descent (PGD), Carlini & Wagner algorithms, black-box or white-box attacks, Byzantine attacks, data poisoning, model extraction, model reverse engineering, model stealing.

G06N 3/096

Definition statement

This place covers:

Techniques that store knowledge gained while solving a given problem or task, and reuse the learned model on another problem or task.

Typical examples or features of such learning or training methods: catastrophic forgetting mitigation, continual learning, incremental learning, lifelong learning, knowledge distillation, teacher-student learning, domain adaptations, knowledge transfers, zero-shot learning, one-shot learning, few-shot learning, multitask learning, common representations, joint representations, shared representations.

G06N 3/098

Definition statement

This place covers:

Techniques wherein features of the neural network model itself or its learning/training enable a distributed or parallel implementation.

Typical examples or features of such learning or training methods: decentralised learning, collaborative learning, federated averaging (FedAvg), parallel gradient ascent or descent, Downpour stochastic gradient descent (D-SGD), subnet training, DistBelief, data parallelism, model parallelism, parameter server, model replicas, data shards, cloud-based learning, client/server-based learning, edge machine learning, MapReduce for machine learning.

G06N 3/0985

Definition statement

This place covers:

Process of finding the right combination of hyperparameter values to achieve maximum performance on the data in a reasonable amount of time.

Learning algorithms that learn from other learning algorithms. For example, meta-data associated with learning techniques are input to another (meta-)learner in order to improve their performance or even induce/learn the (meta-)learning itself.

Typical examples or features of such learning or training methods: automated machine learning (AutoML), neural architecture search (NAS), Bayesian optimisation, algorithm selection, end-to-end learning.

G06N 5/025

Definition statement

This place covers:

Knowledge-based models based on symbolic or knowledge representations including rules (causal, logic, propositional, temporal, if-then-else or antecedent-consequent), or knowledge engineering/acquisition by extracting rules. For example, rule extraction, rule induction, rule elicitation, rule maintenance, rule engines, Apriori algorithm, frequent pattern or itemset mining, association rules.

2. A. DEFINITIONS (modified)

G06N 3/004

Definition statement

Replace: The existing Definition statement text with the following updated text.

Artificial or synthetic life forms that are based on models of or are inspired by natural life forms but are actually implemented or controlled by computing arrangements.

Typical examples of artificial life (Alife) models: agent-based models, multi-agent systems, cellular automata, collective behaviours, self-organised systems, swarm intelligence.

G06N 3/006

Definition statement

Replace: The existing Definition statement text with the following updated text.

Software simulations on computing arrangements of systems exhibiting behaviours normally ascribed to life forms.

Typical examples of Alife based on simulated virtual life forms: ant colony optimisation (ACO), ant clustering, Ant-Miner, artificial bee colonies (ABC), artificial immune systems (AIS), firefly algorithms, particle swarm clustering, particle swarm classification, autonomous agents or bots, intelligent agents or bots, learning agents or bots, smart agents or bots, metaverse, virtual reality, virtual world, virtual society, virtual creatures.

References

Application-oriented references

Examples of places where the subject matter of this place is covered when specially adapted, used for a particular purpose, or incorporated in a larger system:

Insert: The following new reference in the Application-oriented references table.

Computer-aided design [CAD] for design optimisation, verification or simulation	G06F 30/20

Informative references

Attention is drawn to the following places, which may be of interest for search:

Replace: The existing symbol in this row with the following updated symbol.

Collaborative systems - Groupware	G06Q 10/101

G06N 3/008

Definition statement

Replace: The existing Definition statement text with the following updated text.

Computing arrangements emulating/simulating existing biological life forms mainly implemented as physical robots in the form of animals (pets) or humans (humanoids or androids). These physical entities can be standalone or work in groups/swarms (e.g. Robocup team of robotic football players).

Typical examples of Alife based on physical entities: humanoids, androids, robotic pets, autonomous robots, intelligent robots, learning robots, smart robots, behaviour-based robotics.

References

Application-oriented references

Examples of places where the subject matter of this place is covered when specially adapted, used for a particular purpose, or incorporated in a larger system:

Insert: The following two new references in the Application-oriented references table.

Programme-controlled manipulators	B25J 9/00
Total factory control	G05B 19/418

G06N 3/042

Definition statement

Replace: The existing Definition statement text with the following updated text.

Combinations of neural networks and knowledge-based models, in particular knowledge representation and reasoning (KRR) models such as expert systems. This place contains documents where knowledge-based models and neural networks work together on the same level and also where knowledge-based models are used to represent, approximate, construct, augment, support, explain or control a neural network.

Typical examples of such neural network models: rule-based networks, graph networks, hybrid networks, surrogate networks, response surface networks, physics-augmented networks, neural ordinary differential equations, neural tensor networks, symbolic networks, neuro-symbolic systems.

Insert: The following new Relationships with other classification places section.

Relationships with other classification places

Where the knowledge-based models are within neural networks, classification should be made in group G06N 3/042 only.

References

Informative references

Attention is drawn to the following places, which may be of interest for search:

Insert: The following new reference in the Informative references table.

Knowledge representation; Symbolic representation	G06N 5/02

G06N 3/043

Definition statement

Replace: The existing Definition statement text with the following updated text.

Combinations of neural networks and fuzzy logic or inference. This place contains documents where fuzzy logic/inference and neural networks work together on the same level and also where fuzzy logic/inference is used to represent, approximate, construct, augment, support, explain or control a neural network.

Insert: The following new Relationships with other classification places section.

Relationships with other classification places

Where the fuzzy-based models are within neural networks, classification should be made in group G06N 3/043 only.

References

Informative references

Attention is drawn to the following places, which may be of interest for search:

Replace: The existing Informative references table with the following updated table.

Fuzzy inferencing	G06N 5/048
Computing arrangements using fuzzy logic	G06N 7/02

Glossary of terms

In this place, the following terms or expressions are used with the meaning indicated:

Replace: The existing Glossary of terms table with the following updated table.

ANFIS	Adaptive neuro-fuzzy inference systems

G06N 3/044

Definition statement

Replace: The existing Definition statement text with the following updated text.

Neural networks where connections between nodes can create a cycle, allowing output from some nodes to affect subsequent input to the same nodes, and exhibiting temporal dynamic behaviours.

Typical examples of such neural network models: associative memories, feedback networks, Elman networks, reservoir computing, echo state networks (ESN), liquid state machines (LSM), Boltzmann machines.

G06N 3/045

Definition statement

Replace: The existing Definition statement text with the following updated text.

> Architectures wherein multiple neural networks are connected in parallel or in series, including modular architectures wherein components/modules may be represented as neural networks. The neural networks can cooperate on the same level or one neural network can represent, approximate, construct, augment, support, explain or control another neural network.
>
> Parallel neural networks can also be used for fault tolerance when connecting to a voting system.
>
> Several neural networks can also be trained in different ways or with different training examples and then combined in parallel to increase the reliability or accuracy.
>
> Typical examples of such neural network models: multiple neural networks, hierarchical networks, pyramidal networks, modular networks, neural network ensembles, stacked networks, cascaded networks, mixture of expert (MoE) networks, hierarchical temporal memories (HTM), cortical learning algorithms (CLA), dueling networks, adversarial networks, Siamese networks, triplet networks, latent space models, network embeddings, memory-augmented neural networks (MANN), memory networks, neural Turing machines (NTM), differentiable neural computers (DNC), networks with attention mechanisms, transformers, bidirectional encoder representations from transformers (BERT), generative pre-trained transformers (GPT-2, GPT-3), distributed neural networks.

References

Insert: The following new Informative references section.

Informative references

Attention is drawn to the following places, which may be of interest for search:

Ensemble learning	G06N 20/20

G06N 3/047

Definition statement

Replace: The existing Definition statement text with the following updated text.

Neural networks having as special feature that the neurons individually, or the weights connecting the neurons, or the architecture as a whole, have a probabilistic, stochastic or statistical aspect.

Typical examples of such neural network models: Bayesian neural networks, Boltzmann machines, probabilistic RAM (pRAM).

References

Informative references

Attention is drawn to the following places, which may be of interest for search:

Replace: The text in the second row of the Informative references table with the following updated text.

Neural networks based on fuzzy logic, fuzzy membership or fuzzy inference, e.g. adaptive neuro-fuzzy inference systems [ANFIS]	G06N 3/043

G06N 3/048

Definition statement

Replace: The existing Definition statement text with the following updated text.

Function that converts a weighted sum of input data into an output signal. In artificial neural networks, the size of the weighted sum for the previous layer determines whether it is active or not.

Typical examples of activation functions: sigmoid, logistic, hyperbolic tangent (tanh), step function, Heaviside, thresholding, softmax, maxout, rectified linear unit (ReLU), piecewise linear activation functions, radial basis functions (RBF), Gaussian error linear unit (GELU), exponential linear unit (ELU), ridge functions, fold functions.

G06N 3/049

Definition statement

Replace: The existing Definition statement text with the following updated text.

Neurons or neural networks having a temporal aspect, e.g. spiking neurons or neural networks where the time-like dynamics are a specific aspect of the invention. This can be in digital, but is often in analogue technology. These neurons are meant to be a more realistic simulation of real biological neurons.

Typical examples or features of such neural network models: integrate-and-fire (IF) neurons, resonate-and-fire neurons, FitzHugh-Nagumo model, Hodgkin-Huxley model, Izhikevich models, conductance-based models, compartmental models, multi-compartment models, tempotron, time delay networks, address event representations (AER), neuromorphic behaviours, spike-timing-dependent plasticity (STDP), bursting, firing, spiking, population coding, rate coding, temporal coding.

G06N 3/065

Definition statement

Replace: The existing Definition statement text with the following updated text.

Neurons or interconnections implemented using analogue electronics, including mixed-signal or hybrid analogue-digital electronics.

Typical examples of such neural network realisations: neuromorphic chips, neuromorphic circuits, neuromorphic systems, neurons or synapses implemented with memristors, with memristive systems, or with non-volatile memories (NVM) typically arranged in arrays, e.g. in crossbar (XB) arrays.

References

Informative references

Attention is drawn to the following places, which may be of interest for search:

Replace: The existing Informative references table with the following updated table.

Analogue computers in general	G06G 7/00
Hybrid computing arrangements	G06J 1/00

G06N 3/08

Insert: The following new Relationships with other classification places section.

Relationships with other classification places

Where the machine learning relates to learning methods within neural networks, classification should be made in group G06N 3/08 only.

G06N 3/082

Definition statement

Replace: The existing Definition statement text with the following updated text.

During the learning or training process of the neural network, not only are the weights of the synapses changed, but also is the architecture of the neural network changed, even temporarily. This can involve adding/deleting/silencing neurons or adding/deleting/silencing connections between the neurons.

When during the training process it becomes clear that the size/capacity of the neural network is not sufficient, additional neurons or connections can be added to the network after which the training can resume. When it is found that certain neurons are not used or have no influence, they can be removed (pruning). Such modifications can be part of a search for optimal architectures (neural architecture search). Neurons or connections can also be silenced for regularisation (dropout/dropconnect) and improving the network’s generalisation.

G06N 3/084

Definition statement

Replace: The existing Definition statement text with the following updated text.

Training method whereby the synapses of the neurons are adapted depending on the difference between the actual output of the neural network and the wanted output. This difference is used to adapt the weights of the synapses with a mathematical method that back-propagates from the higher layers to the lower layers of the neural network. Mainly used in multilayer neural networks.

Typical examples of such learning or training methods: feedback alignment, automatic differentiation, backprop, error-backpropagation, backward propagation of errors based on gradient ascent or gradient descent (e.g. stochastic or minibatch gradient descent, Adagrad, Adam, RMSprop).

G06N 3/086

Definition statement

Replace: The existing Definition statement text with the following updated text.

The use of evolutionary algorithms for creating an optimally functioning neural network, such as evolutionary programming, genetic algorithms, genetic programming, evolution strategies, etc.

References

Informative references

Attention is drawn to the following places, which may be of interest for search:

Replace: The existing Informative references table with the following updated table.

Evolutionary algorithms, e.g. genetic algorithms or genetic programming	G06N 3/126

G06N 3/088

Definition statement

Replace: The existing Definition statement text with the following updated text.

Learning or training without direct supervision from unlabelled data. Neural networks are created, and then it is observed how they function in the real world. As a result of global functioning, the neural network is further adapted. No sets of ground truth data are necessary, and input data are, for example, clustered.

Typical examples of non-supervised or unsupervised methods: competitive learning, self-organising maps (SOM), self-organising feature maps (SOFM), Kohonen maps, topological maps, neural gas, neural network clustering, anomaly detection, contrastive divergence algorithms, expectation-maximisation (EM), spike-timing-dependent plasticity (STDP), variational inference, wake-sleep algorithms, Hebbian learning, Hebb's rule, Oja's rule, BCM rule.

G06N 3/123

Definition statement

Replace: The existing Definition statement text with the following updated text.

Information processing using DNA, whereby a computational problem (e.g. optimisation) is represented with, or encoded on DNA molecules which are manipulated in such a way that at least one DNA molecule is produced that represents a solution to the problem.

G06N 3/126

Definition statement

Replace: The existing Definition statement text with the following updated text.

Software simulations using the principles evolution as exhibited in real biological systems. For example, genetic algorithms (GA) involve the creation of a number of possible solutions (chromosomes or individuals), testing the different solutions by evaluating a fitness, performance or score function (representing an optimisation problem such as classification, clustering or regression), selecting the best performing ones, starting from these to create a new set of possible solutions using reproduction and mutation, and reiterating until an optimal or sufficiently performing solution is found.

Typical examples of evolutionary algorithms (EA): gene expression programming (GEP), evolutionary programming (EP), memetic algorithms (MA), evolution strategies (ES), covariance matrix adaptation evolutionary strategies (CMA-ES), Darwinistic programming, differential evolution (DE), estimation of distribution algorithms (EDA), probabilistic model-building genetic algorithms (PMBGA), co-evolution, learning classifier systems (LCS), niche-based EA, island-based EA, diffusion grid EA, cellular EA, parallel EA, distributed EA, fine-grained EA,, coarse-grained EA, multi-objective EA (MOEA), non-dominated sorting GA (NSGA).

Relationships with other classification places

Replace: The existing Relationships text with the following updated text.

Classification in this group is not expected when evolutionary algorithms are used in training neural networks. Applications of whatever sort just using evolutionary algorithms with no description of the evolutionary algorithm itself are to be classified in the relevant application field only.

References

Application-oriented references

Examples of places where the subject matter of this place is covered when specially adapted, used for a particular purpose, or incorporated in a larger system:

Replace: The term “Genetic” with “Evolutionary” in the Application-oriented references table.

Evolutionary algorithms used in training of neural networks	G06N 3/086

G06N 5/01

Definition statement

Replace: The existing Definition statement text with the following updated text.

Techniques for searching or exploring the solution space of an optimisation problem, such as dynamic programming, branch-and-bound, breadth-first search, depth-first search, shortest path algorithms, techniques based on tree or graph representations (e.g. tree- or graph-traversal, Monte Carlo tree search), first-order logic (e.g. automatic theorem proving), heuristics or models based on empirical knowledge. The optimisation problem is typically defined by one or more objectives or constraints. The outcome may represent an optimal solution, or an indication that the problem can be solved or that its solutions can be verified. Such techniques are normally used when classic methods fail to find an exact solution in a short time.

Typical examples of such techniques: annealing techniques, Monte Carlo search techniques, adaptive search techniques, exploration-exploitation techniques, constraint solvers, constraint optimisations, empirical optimisations, replica methods, predicate logic, iterative dichotomiser 3 (ID3), C4.5 algorithms, classification and regression trees (CART), decision trees, isolation or random forests, good old-fashioned artificial intelligence (GOFAI) techniques.

References

Application-oriented references

Examples of places where the subject matter of this place is covered when specially adapted, used for a particular purpose, or incorporated in a larger system:

Replace: The existing Application-oriented references table with the following updated table.

Video games	A63F 13/00
Complex mathematical operations for solving equations	G06F 17/11
Computer-aided design [CAD] for design optimisation, verification or simulation	G06F 30/20
Forecasting or optimisation specifically adapted for administration or management purposes	G06Q 10/04
ICT specially adapted for medical diagnosis, medical simulation or medical data mining; ICT specially adapted for detecting, monitoring or modelling epidemics or pandemics	G16H 50/00

G06N 5/02

Definition statement

Replace: The existing Definition statement text with the following updated text.

Knowledge-based models based on specific symbolic or knowledge representations, knowledge engineering, or knowledge acquisition. Typical examples of representations: knowledge bases, knowledge graphs, knowledge repositories, knowledge corpus, predicates, ontologies, taxonomies, semantic networks and other graph-based representations of knowledge.

Insert: The following new Relationships with other classification places section.

Relationships with other classification places

Where the knowledge-based models are within neural networks, classification should be made in group G06N 3/042 only.

References

Informative references

Attention is drawn to the following places, which may be of interest for search:

Replace: The term “indexing” with “Indexing” in the Informative references table.

Indexing in information and retrieval	G06F 16/00

G06N 5/043

Definition statement

Replace: The existing Definition statement text with the following updated text.

Expert systems implemented in distributed programming units or multiple interacting intelligent autonomous components, for example, multi-agent systems.

G06N 5/045

Definition statement

Replace: The existing Definition statement text with the following updated text.

Inference or reasoning model that provides or supports explanations or interpretations of the inferences or reasoning to the user in the context of diagnostic or decision support.

Synonyms and Keywords

In patent documents, the following words/expressions are often used as synonyms:

Replace: The existing Synonyms and Keywords text with the following updated text.

- "explanation", “anomaly”, "decision", "diagnostic", "fault", "abnormal" and "alarm"

G06N 5/046

Definition statement

Replace: The existing Definition statement text with the following updated text.

> Inference or reasoning that starts with the available data and makes inferences to derive more data. The inferences are performed forwards towards a goal by repetitive application of the modus ponens.

G06N 5/048

Definition statement

Replace: The existing Definition statement text with the following updated text.

> Transformation of exact inputs in fuzzy inputs with membership functions (fuzzification). The fuzzified inputs are processed in a fuzzy inference machine with fuzzy if-then rules. Depending on the degree of membership, several rules are fired in parallel. The consequents of each rule are aggregated into fuzzy outputs which are de-fuzzified or not de-fuzzified.

Insert: The following new Relationships with other classification places section.

> **Relationships with other classification places**
> Where the fuzzy-based models are within neural networks, classification should be made in group G06N 3/043 only.

References

Informative references

Attention is drawn to the following places, which may be of interest for search:

Replace: The existing Informative references table with the following updated table.

Computing arrangements using fuzzy logic	G06N 7/02

G06N 7/01

Definition statement

Replace: The existing Definition statement text with the following updated text.

Probabilistic graphical model (PGM) is a probabilistic model for which a graph expresses the conditional dependence structure between random variables, such as belief networks, Bayesian networks, Markov models, Markov decision process (MDP), conditional random fields (CRF), Markov chain Monte Carlo (MCMC). Typical examples of graphical models: structured probabilistic models, Bayes networks, directed acyclic graph models, belief propagation, influence diagrams, latent Dirichlet allocation (LDA), Bayes classifiers, Bayesian optimisation, Ising models, Pott models, spin-glass models, Markov chains, Markov networks, Markov random fields (MRF).

Relationships with other classification places

Replace: The existing Relationships text with the following updated text.

Classification in this group is not expected when probabilistic graphical models are used in neural networks (e.g. Boltzmann machines).

Applications of whatever sort just using Bayesian or Markov models with no description of the Bayesian or Markov model itself are to be classified in the relevant application field.

Learning of unknown parameters of the network should also be classified in G06N 20/00.

References

Application-oriented references

Examples of places where the subject matter of this place is covered when specially adapted, used for a particular purpose, or incorporated in a larger system:

Replace: The existing Application-oriented references table with the following updated table.

Video games	A63F 13/00
Digital data processing	G06F
Information retrieval	G06F 16/00
Pattern recognition	G06F 18/00
Classification of content in image-based pattern recognition	G06V 30/413
Speech recognition	G10L 15/00

Informative references

Attention is drawn to the following places, which may be of interest for search:

Replace: The existing Informative references table with the following updated table.

Recurrent networks, e.g. Hopfield networks	G06N 3/044
Neural networks having a probabilistic aspect	G06N 3/047
Generative networks	G06N 3/0475

G06N 20/00

Insert: The following new Relationships with other classification places section.

Relationships with other classification places

Where the machine learning relates to learning methods within neural networks, classification should be made in group G06N 3/08 only.

References

Informative references

Attention is drawn to the following places, which may be of interest for search:

Replace: The existing Informative references table with the following updated table.

Adaptive control systems	G05B 13/00
Computing arrangements using neural networks	G06N 3/02
Computing arrangements using knowledge-based models	G06N 5/00
Computing arrangements using fuzzy logic	G06N 7/02
Image processing using neural networks	G06T 1/20
Image or video recognition or understanding using machine learning	G06V10/70, G06V 30/194
Speech recognition using artificial neural networks	G10L 15/16