(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
10 April 2025 (10.04.2025)

WIPO | PCT

(10) International Publication Number
WO 2025/075604 A1

(51) International Patent Classification:
G06F 18/2321 (2023.01) *G06N 3/088* (2023.01)
G06N 3/045 (2023.01) *G06F 18/23* (2023.01)
G06N 3/0475 (2023.01) *G06F 18/21* (2023.01)

(21) International Application Number:
PCT/US2023/034282

(22) International Filing Date:
02 October 2023 (02.10.2023)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant: **GOOGLE LLC** [US/US]; 1600 Amphitheatre Parkway, Mountain View, California 94043 (US).

(72) Inventors: **PÉRAUD, Soyoung**; 1600 Amphitheatre Parkway, Mountain View, California 94043 (US). **CHU, Wen-Sheng**; 1600 Amphitheatre Parkway, Mountain View, California 94043 (US). **LIU, Yaojie**; 1600 Amphitheatre Parkway, Mountain View, California 94043 (US).

(74) Agent: **PORTNOV, Michael**; Fish & Richardson P.C., PO Box 1022, Minneapolis, Minnesota 55440 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CV, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, MG, MK, MN, MU, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, CV, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SC, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, ME, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report (Art. 21(3))*

(54) Title: IDENTIFYING OUT-OF-DISTRIBUTION INPUTS USING GENERATIVE MODELS

(57) Abstract: Methods, systems, and apparatus, including computer programs encoded on a computer storage medium, for identifying out-of-distribution inputs to a machine learning model. In one aspect, a method comprises receiving a network input, processing the network input using a pre-trained model composed of a plurality of layers, generating features of the network input based on the layer outputs, and generating an out-of-distribution score for the network input that represents the deviation of the network input from a distribution determined by a generative model configured to characterize a particular distribution of a subset of training data.

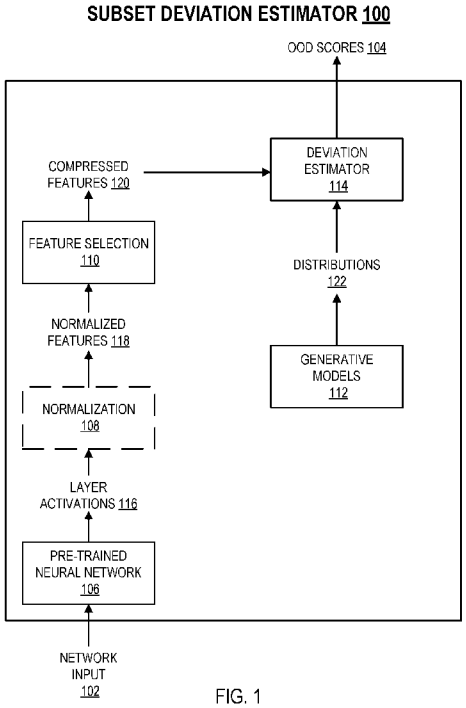


FIG. 1

WO 2025/075604 A1

IDENTIFYING OUT-OF-DISTRIBUTION INPUTS USING GENERATIVE MODELS

BACKGROUND

[0001] This specification relates to processing data using machine learning models.

[0002] Machine learning models receive an input and generate an output, e.g., a predicted output, based on the received input. Some machine learning models are parametric models and generate the output based on the received input and on values of the parameters of the model.

[0003] Some machine learning models are deep models that employ multiple layers of models to generate an output for a received input. For example, a deep neural network is a deep machine learning model that includes an output layer and one or more hidden layers that each apply a non-linear transformation to a received input to generate an output.

SUMMARY

[0004] This specification describes a system implemented as computer programs on one or more computers in one or more locations that can identify out-of-distribution inputs to a machine learning model.

[0005] According to a first aspect there is provided a method, the method comprising: receiving a network input; processing the network input using a pre-trained neural network composed of a plurality of layers; generating, from the outputs of one or more of the layers, features of the network input; and generating an out-of-distribution (OOD) score for the network input that represents the deviation of the network input from a distribution determined by a generative model configured to characterize a particular distribution of a subset of training data.

[0006] Particular embodiments of the subject matter described in this specification can be implemented so as to realize one or more of the following advantages.

[0007] By utilizing generative models in an unsupervised manner, the described systems only require in-distribution training data, such as the data used to train other neural networks in a processing system. Unlike conventional methods of out-of-distribution characterization, the described systems do not require labeled out-of-distribution training samples and can therefore be used in settings where realistic out-of-distribution examples are unavailable. By enabling an out-of-distribution characterization of input data as using multiple OOD scores, the described systems provide a tunable method for accurate out-of-distribution characterization. By employing a process of feature selection to reduce the dimensionality of the features used to compute the OOD scores, the described systems can perform out-of-distribution

characterization more efficiently and more reliably than conventional methods. The described systems therefore enable an efficient method for out-of-distribution characterization that learns and represents the distributions in question without requiring the collection and labeling of data outside those distributions.

[0008] The details of one or more embodiments of the subject matter of this specification are set forth in the accompanying drawings and the description below. Other features, aspects, and advantages of the subject matter will become apparent from the description, the drawings, and the claims.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009] FIG. 1 shows an example subset deviation estimation system for quantifying the extent to which an input falls outside a learned distribution.

[0010] FIG. 2 illustrates an example of calculating an OOD score based on a Gaussian Mixture Model distribution with two components using the Mahalanobis distance.

[0011] FIG. 3 shows an example out-of-distribution (OOD) classification system.

[0012] FIG. 4 illustrates an example of refining an initial classification using an OOD classification system.

[0013] FIG. 5 is a flow diagram of an example process for producing a refined out-of-distribution classification.

[0014] Like reference numbers and designations in the various drawings indicate like elements.

DETAILED DESCRIPTION

[0015] These features and other features are described in more detail below.

[0016] FIG. 1 shows an example subset deviation estimator system 100. The system 100 is an example of a system implemented as computer programs on one or more computers in one or more locations in which the systems, components, and techniques described below are implemented.

[0017] The system 100 is configured to process a network input 102 to produce a set of one or more out-of-distribution (OOD) scores 104 that quantify how closely the network input 102 aligns with chosen subsets of data.

[0018] Each OOD score 104 represents a computed deviation between the network input 102 and a distribution characterizing a particular subset of data represented by the system 100.

[0019] Once the system 100 has generated the OOD scores 104, the system 100 can use the set of OOD scores for any of a variety of purposes.

[0020] For example, the system 100 can use the OOD scores 104 produced to characterize the potential membership of the network input 102 to the data subsets represented within the system 100. The OOD scores 104 can therefore be used to make decisions regarding the network input 102 based on how much the network input 102 is out-of-distribution from the data subsets represented within the system 100.

[0021] As one example, it is known that neural networks can produce unreliable or spurious outputs when applied to data falling outside the distributions of their training data. It may be critical to avoid providing outputs generated by a particular neural network, e.g. the pre-trained neural network 106, from data that is OOD from the particular network's training set. In this case, the system 100 can be configured to calculate OOD scores 104 with respect to subsets of the particular network's training data indicating the reliability of the particular neural network for processing the network input 102.

[0022] The particular neural network can be trained to perform any kind of machine learning task, i.e., can be configured to receive any kind of digital data input and to generate any kind of score, classification, regression output, or generative output based on the network input 102.

[0023] In some cases, the particular neural network is a neural network that is configured to perform an image processing task, i.e., receive an input image as the network input 102 and to process the input image, i.e., process the intensity values of the pixels of the input image, to generate a network output for the input image. For example, the task may be image classification and the output generated by the particular neural network for a given image may be scores for each of a set of object categories, with each score representing an estimated likelihood that the image contains an image of an object belonging to the category. As yet another example, the task can be object detection and the output generated by the particular neural network can identify locations in the input image at which particular types of objects are depicted. As yet another example, the task can be image segmentation and the output generated by the particular neural network can assign each pixel of the input image to a category from a set of categories.

[0024] As another example, if the network inputs 102 to the particular neural network are Internet resources (e.g., web pages), documents, or portions of documents or features extracted from Internet resources, documents, or portions of documents, the task can be to classify the resource or document, i.e., the output generated by the particular for a given Internet resource, document, or portion of a document may be a score for each of a set of topics, with each score representing an estimated likelihood that the Internet resource, document, or document portion is about the topic.

[0025] As another example, if the network inputs 102 to the particular neural network are features of an impression context for a given advertisement, the output generated by the particular neural network may be a score that represents an estimated likelihood that the given advertisement will be clicked on.

[0026] As another example, if the network inputs 102 to the particular neural network are features of a personalized recommendation for a user, e.g., features characterizing the context for the recommendation, e.g., features characterizing previous actions taken by the user, the output generated by the particular neural network may be a score for each of a set of content items, with each score representing an estimated likelihood that the user will respond favorably to being recommended the content item.

[0027] As another example, the task may be an audio processing task. As another example, the task may be a keyword spotting task where, if the network input 102 to the particular neural network is a sequence representing a spoken utterance, the output generated by the particular neural network can indicate whether a particular word or phrase ("hotword") was spoken in the utterance. As another example, if the network input 102 to the particular neural network is a sequence representing a spoken utterance, the output generated by the neural network can identify the natural language in which the utterance was spoken.

[0028] As another example, the task can be a natural language processing or understanding task, e.g., an entailment task, a textual similarity task, a sentiment task, a grammaticality task, and so on, that operates on a sequence of text in some natural language.

[0029] As another example, the task can be a health prediction task, where the network input 102 is electronic health record data for a patient and the output is a prediction that is relevant to the future health of the patient, e.g., a predicted treatment that should be prescribed to the patient, the likelihood that an adverse health event will occur to the patient, or a predicted diagnosis for the patient.

[0030] In some cases, the particular neural network can be a conditional generative model that is configured to perform a conditional generation task and the network input 102 may be a prompt to be provided to the model for generating samples. The generative model might produce unreliable or unrealistic samples when used with a prompt far removed from the set of prompts used to train the generative model. By representing subsets of the prompts used to train the generative model, the system 100 can produce OOD scores 104 that indicate whether the model's samples based on a given prompt are likely unreliable or unrealistic.

[0031] As an example, the conditional generation task can be image generation and the output generated by the particular neural network can be an image conditioned on the prompt.

[0032] As another example, if the prompt to the particular neural network is a sequence of text in one language, the output generated by the particular neural network may be text in another language.

[0033] As another example, if the prompt to the particular neural network is a sequence representing a spoken utterance, the output generated by the particular neural network may be a text transcript for the utterance.

[0034] As another example, the conditional generation task can be a text to speech task, where the prompt is text in a natural language or features of text in a natural language and the network output is a spectrogram or other data defining audio of the text being spoken in the natural language.

[0035] As another example, the conditional generation task can be a natural language processing task, e.g., a paraphrase task, a sentence completion task, and so on, where the prompts are text in some natural language.

[0036] As another example, the conditional generation task can be a text generation task, where the prompt is a question or the history of a conversation, e.g., with a user, and the network output is an appropriate response.

[0037] As another example, the conditional generation task can be an agent control task, where the prompt is an observation characterizing the state of an environment and the output defines an action to be performed by the agent in response to the observation. The agent can be, e.g., a real-world or simulated robot, a control system for an industrial facility, or a control system that controls a different kind of agent.

[0038] Another related use of the OOD scores 104 is as part of a distribution-aware refinement of an initial classification of the network input 102. A classification of the network input 102 is a characterization of the membership of the network input 102 to one or more of multiple data subsets called classes. A classification neural network can provide a classification of the network input 102 by producing a label encoding which one or more classes are assigned to the network input 102 or by producing a set of numerical values, e.g. logits, probabilities, or likelihoods, characterizing the probability of membership of the network input 102 to each class. An initial classification can be refined by classifying the input into the same classes with a more accurate label or class probability assignment. An initial classification can also be refined by classifying the input into an expanded or more finely sub-divided set of classes. By configuring the system 100 to represent subsets of data relevant to the refined classification, the resulting OOD scores 104 can provide useful information for producing a refined classification that can indicate that pre-determined out-of-distribution actions, such as rejecting

the input 102 or alerting to the user that system 100 is uncertain of the input 102, are appropriate. As an example, the OOD scores 104 may be used by a refined classifier that adds an additional "uncertain" class to the set of initial classifications indicating that the system is uncertain of the initial classification. By processing the OOD scores 104, the refined classifier can additionally provide information characterizing the uncertainty assigned to the initial classification. As another example, a refined classifier can include a class for exceedingly rare events and can use the OOD scores 104 to identify examples of these rare events that are otherwise difficult to include within sets of training data.

[0039] The network input 102 can be represented as an ordered collection of numerical values, e.g. as a one-dimensional (1-D), two-dimensional (2-D), three-dimensional (3-D), or *N*-dimensional (N-D) array of numerical values (where *N* is a positive integer). In some implementations, the network input 102 can be data representing an image.

[0040] The system 100 includes a pre-trained neural network 106. To generate the set of OOD scores 104, the system 100 processes the network input 102 using the pre-trained neural network 106 to produce layer activations 116.

[0041] The pre-trained neural network 106 includes one or more network layers used by the network 106 to process the network input 102, with each layer processing a layer input and producing a layer output.

[0042] The pre-trained neural network 106 can be configured to perform any processing task on the network input 102 and can have any appropriate neural network architecture enabling it to perform that processing task. In implementations where the network input 102 represents an image, the pre-trained neural network 106 can be an image processing network, with a suitably specialized architecture, e.g. a convolutional neural network architecture or a Vision Transformer neural network architecture.

[0043] The layer activations 116 are a pre-determined subset of the layer outputs resulting from the pre-trained neural network 106 processing the network input 102. The pre-trained neural network 106 contains multiple hidden layers, that can include processing layers (e.g., linear or convolutional layers), activation layers (e.g., softmax or ReLU layers), and normalization layers (e.g., batch norm or layer norm layers). Neural networks are typically constructed by alternating various processing layers with various activation and normalization layers. Although the layer activations 116 can be the layer outputs from all the layers within the pre-trained neural network 106, using all of the layer outputs can result in the layer activations 116 containing many redundant values that require additional computational processing without a substantial increase in accuracy in generating the OOD scores 104. To reduce the

computational burden of processing the layer activations 116, the layer activations 116 can be a subset of the layer outputs. For example, if the pre-trained neural network 106 is constructed by alternating linear layers and normalization layers, the layer activations 116 can be the outputs of all or some of the normalization layers. As another example, if the pre-trained neural network 106 is constructed by alternating convolutional layers and pooling layers, the layer activations 116 can be the outputs of all or some of the pooling layers.

[0044] Optionally, the system 100 can then process the layer activations 116 by applying a normalization 108 to produce normalized features 118.

[0045] The normalization 108 is a transformation to the layer activations 116 such that the normalized features 118 follow particular pre-determined statistical properties. For example, the normalization 108 can be the transformation of subtracting the means of the layer activations 116 and dividing by the standard deviations of the layer activations 116 so as to produce normalized features 118 having zero mean and unit variance. In implementations where the normalization 108 is not applied, the normalized features 118 are the original layer activations 116.

[0046] The system 100 can apply a process of feature selection 110 to the normalized features 118 to produce compressed features 120.

[0047] The compressed features 120 are referred to as "compressed" because the compressed features 120 can have a smaller dimensionality relative to the dimensionality of the normalized features 118. The compression performed by the feature selection 116 removes redundant values remaining within the normalized features 118, which can lead to numerical instability in later probabilistic modeling and reduce the reliability of the system 100. While selecting an appropriate subset of layer outputs to form the layer activations 116 can avoid introducing values from redundant layers, the feature selection 116 is able to remove any remaining redundant values present within the layer outputs. The feature selection 110 can include a set of feature selection parameters that determine aspects of the feature selection 110. These feature selection parameters can be determined by any appropriate procedure optimizing how well the resulting compressed features 120 summarize the network input 102 for the purposes of determining membership within the subsets of data considered by the system 100.

[0048] In some implementations, the feature selection 110 is the application of principal components analysis (PCA) to the normalized features 118. In implementations where the feature selection 110 is the application of PCA, the feature selection parameters can include the output dimension of the compressed features 120. This output dimension can be determined

based on the proportion of variance of the normalized features 118 that is explained by the resulting compressed features 120.

[0049] The system 100 includes one or more generative models 112. The generative models 112 are configured to model particular subsets of data and include generative model parameters that determine the modeled distributions 122. The generative model parameters can be determined by any suitable procedure that ensures the generative models 112 characterize the desired data subsets. For example, the generative model parameters may be optimized by training the generative models using sets of training data that are distributed according to distributions of the compressed features 120. Here, the modeled distributions 122 are described as being composed of one or more component distributions.

[0050] The generative models 112 can be generative neural networks with architectures enabling the computation of an exact or approximate likelihood of the modeled data. In some implementations, the generative models 112 are Gaussian Mixture Models (GMMs) that model distributions having Gaussian component distributions.

[0051] In implementations where network input 102 can be classified among multiple classes, each of the generative models 112 can be responsible for modeling the distribution of a corresponding class label. As an example, if network input 102 comprises images that can be classified as either "animal" and "not animal" images, the generative models 112 can include a model of the distribution of "animal" images and a model of the distribution of "not animal" images. The component distributions represented by each of the generative models 112 can then characterize the distributions of clusters of data within each of the class subsets. In the same example, the generative model for "animal" images can include a component distribution modeling images of dogs and another component distribution modeling images of cats, while the generative model for "not animal" images can include a component distribution modeling images of plants and another component distribution modeling images of buildings.

[0052] The system 100 includes a deviation estimator system 114 can process the compressed features 120 and modeled distributions 122 generated by the one or more generative models 112 using the system 114 to produce the OOD scores 104 that quantify the alignment of the compressed features 120 with the distributions 122.

[0053] The system 114 is configured to produce OOD scores 104 by computing component deviations between the compressed features 120 and each component of the distributions 122. Each component deviation characterizes the likelihood of the compressed features 120 within a corresponding component distribution described by one of the generative models 112. In the above example, component deviations are computed between the compressed features 120 and

the component distributions of dog and cat images that characterize the respective likelihoods of the network input 102 as an image of a dog or of a cat.

[0054] In some implementations, the component deviations are calculated as the Mahalanobis distances between the compressed features 120 and each of the component distributions. In implementations where the Mahalanobis distances are calculated, the component deviation, $\Delta_{i,j}$, between the compressed features, $\vec{z}$, and the j-th component of the i-th model, having mean $\vec{\mu}_{i,j}$ and covariance $\Sigma_{i,j}$, is determined by:

$$\Delta_{i,j} = \sqrt{\left(\vec{z} - \vec{\mu}_{i,j}\right)^T \Sigma_{i,j}^{-1} \left(\vec{z} - \vec{\mu}_{i,j}\right)}$$

[0055] In implementations where the component deviation is calculated as the Mahalanobis distance and where the generative models 112 are Gaussian Mixture Models, the system 114 therefore computes each component deviation, $\Delta_{i,j}$, directly from the compressed features 120 and the generative model parameters.

[0056] The system 114 processes the resulting component deviations to produce the output set of OOD scores 104, with each OOD score characterizing the likelihood of the compressed features 120 under a corresponding overall distribution represented by one of the generative models 112. Continuing the above example, the system 114 can combine the dog and cat component deviations to produce an OOD score that characterizes the likelihood of the network input 102 as an image of an animal.

[0057] As one example of how the system 114 can combine the component deviations, the OOD scores 104 can be determined as the minimum of the component deviations for each modeled distribution, with the score for the i-th model, s_i, being determined following:

$$s_i = \min_j \Delta_{i,j}$$

[0058] In the example application where the system 100 is configured to represent subsets of the training data of a particular neural network, the OOD scores 104, along with a predefined threshold τ_i for each OOD score s_i, can indicate whether the particular neural network should be considered unreliable in processing network input 102 when the scores 104 satisfy:

$$\min_i \left(\frac{s_i}{\tau_i}\right) > 1$$

[0059] FIG. 2 illustrates an example of calculating an OOD score 104 based on a Gaussian Mixture Model distribution with two components using the Mahalanobis distance. The system 100 can compute an OOD score for the overall distribution depicted in FIG. 2 by computing component deviations between the compressed features 120 and each of the Gaussian

component distributions 202-A and 202-B. The system 100 can compute these component deviations as the Mahalanobis distances 204-A and 204-B between the compressed features 120 and the component distributions 202-A and 202-B, respectively. Continuing the above example, the distribution depicted in FIG. 2 can be a distribution of "animal" images of the above example, having Gaussian component distributions of dog images, 202-A, and of cat images, 202-B. To generate an OOD score characterizing the likelihood of the compressed features 120 under the distribution of "animal" images, the system 100 can compute the Mahalanobis distances 204-A and 204-B between the compressed features 120 and the component distributions of dog images, 202-A, and of cat images, 202-B. In this example, the Mahalanobis distances 204-A and 204-B characterize the likelihood of the compressed features 120 as a representation of an image of a dog and as a representation of an image of a cat, respectively.

[0060] As described above, a system can process the OOD scores 104 to produce a refined classification of the network input 102 that improves upon an initial classification of the input 102. Such an out-of-distribution classification system is described with reference to FIG. 3.

[0061] FIG. 3 shows an example out-of-distribution (OOD) classification system 300. The OOD classification system 300 is an example of a system implemented as computer programs on one or more computers in one or more locations in which the systems, components, and techniques described below are implemented.

[0062] The OOD classification system 300 is configured to process a network input 102 to generate a refined classification 302, where the refined classification 302 improves upon an initial classification 306 of the network input 102 by explicitly considering how closely the network input 102 agrees with distributions represented within the OOD classification system 300.

[0063] The OOD classification system 300 includes the system 100, a classification neural network 304, and a refinement system 308. The OOD classification system can process the network input 102 using the system 100 to produce OOD scores 104 and can process the network input 102 using the classification neural network 304 to produce an initial classification 306. The OOD classification system can process the initial classification 306 and the OOD scores 104 using the refinement system 308 to produce the refined classification 302.

[0064] The classification network 304 includes a set of classification network parameters and can be any neural network appropriately configured to classify the network input 102 into the set of possible initial classifications 306. The classification network parameters can be

determined by any suitable procedure that ensures the classification network 304 performs the desired initial classification 306. For example, the classification network parameters can be optimized by training the classification network 304 using a set of ground truth classification training data.

[0065] In implementations where the network input 102 represents an image, the classification network 304 can be an image classification network, with a suitably specialized architecture, e.g. a convolutional neural network architecture or a Vision Transformer neural network architecture.

[0066] In some implementations, the classification network 304 is the pre-trained neural network 106 of the system 100.

[0067] The refinement system 308 is configured to produce the refined classification 302 by processing the OOD scores 104 and the initial classification 306 based on a set of refinement system parameters. The refinement system parameters can be determined by any suitable procedure that ensures the refinement system 308 performs the desired refined classification 302.

[0068] In some implementations, the refinement classification 302 may extend the initial classification 306 by adding a class representing that the OOD classification system 300 is uncertain of the initial classification 306 provided by the classification neural network 306. As an example, the refinement system parameters may specify a threshold τ_i for each OOD score s_i and the refinement system 308 may be configured to produce a refined classification 302 determined by:

$$REFINED\ CLASSIFICATION\ 302 = \begin{cases} "UNCERTAIN", & min\left(\frac{s_i}{\tau_i}\right) > 1 \\ INITIAL\ CLASSIFICATION\ 306, & otherwise \end{cases}$$

[0069] In some implementations, the subsets of data represented by system 100 can be the subsets of data corresponding to the classes of the initial classification 306. If the subsets of data represented by system 100 correspond to the classes of the initial classification 306, the OOD scores 104 produced by the system 100 both characterize the likelihood of the network input 102 as a member of each class of the initial classification 306 and can indicate a degree of uncertainty assigned to the initial classification. Thresholding based on the OOD scores 104, such as using the thresholds τ_i in the preceding example, can allow the system 300 to have a tunable and class-specific sensitivity when determining uncertainty in the initial classification 306. Depending on the application, the system 300 can indicate particular courses of action by returning an uncertain refined classification. For example, an uncertain refined classification

can indicate that the network input 102 should be rejected. As another example, an uncertain refined classification can indicate that additional data needs to be collected in order to process network input 102. In other implementations, the system 300 can return the initial classification 306 along with a measure of its uncertainty in the initial classification 306 as derived from the OOD scores 104.

[0070] FIG. 4 illustrates an example of the refinement of an initial classification that can be performed by the OOD classification system 300. In this example, the classification neural network 304 is configured to classify whether an input image 402 is an image of an animal. In this example, the system 100 represents various image distributions 406-A, 406-B, 406-C and so on. It should be noted that, in this example, the data subsets represented within the system 100 do not correspond exactly to the subsets of data comprising the initial classification classes 404 but are instead related. The OOD scores 104 produced by the system 100 include the individual scores 408-A, 408-B, 408-C and so on. Each of these individual scores represents how closely the input image 402 aligns with the data subsets represented by the subset deviation estimator, i.e. the OOD score 408-A represents how closely the input image 402 aligns with the distribution of known images of dogs 406-A. In this example, the refinement system 308 produces a refined classification that either certainly classifies whether the input image 402 is an image of an animal or reports that the system is uncertain about the classification.

[0071] FIG. 5 is a flow diagram of an example process for calculating an OOD score. For convenience, the process 500 will be described as being performed by a system of one or more computers located in one or more locations. For example, a subset deviation estimator, e.g., the system 100 of FIG. 2, appropriately programmed in accordance with this specification, can perform the process 500.

[0072] The system receives a network input (502).

[0073] The system processes the network input using a pre-trained neural network, yielding a number of network layer activations (504).

[0074] The system generates features based on the layer activations of the pre-trained neural network (506).

[0075] In some implementations, the system applies a normalization to the network layer activations. In these implementations, the system applies a transformation to the network layer activations resulting in normalized layer activations with particular pre-determined statistical properties.

[0076] The system estimates the likelihood of the generated features under the distribution of a generative model (508).

[0077] In some implementations, the system may compute one or more numerical deviations between the generated features and the distribution of the generative model. In these implementations, the numerical deviations may not be identical to the likelihood of the generated features but may instead loosely characterize the likelihood of the generated features.

[0078] In some implementations, generative model may be a Gaussian Mixture Model (GMM). In these implementations, the system may compute one or more numerical deviations between the generated features and the distribution of the GMM by computing a deviation between the generated features and each component Gaussian distribution of the GMM. In some implementations, this computed deviation may be the Mahalanobis distance.

[0079] The system generates an OOD score (510). In particular, the generated OOD score summarizes the one or more numerical deviations and the system may compute the OOD score by any appropriate calculation (e.g. a statistical measure such as the average, minimum, or maximum of the deviations).

[0080] This specification uses the term "configured" in connection with systems and computer program components. For a system of one or more computers to be configured to perform particular operations or actions means that the system has installed on it software, firmware, hardware, or a combination of them that in operation cause the system to perform the operations or actions. For one or more computer programs to be configured to perform particular operations or actions means that the one or more programs include instructions that, when executed by data processing apparatus, cause the apparatus to perform the operations or actions.

[0081] Embodiments of the subject matter and the functional operations described in this specification can be implemented in digital electronic circuitry, in tangibly-embodied computer software or firmware, in computer hardware, including the structures disclosed in this specification and their structural equivalents, or in combinations of one or more of them. Embodiments of the subject matter described in this specification can be implemented as one or more computer programs, i.e., one or more modules of computer program instructions encoded on a tangible non-transitory storage medium for execution by, or to control the operation of, data processing apparatus. The computer storage medium can be a machine-readable storage device, a machine-readable storage substrate, a random or serial access memory device, or a combination of one or more of them. Alternatively or in addition, the program instructions can be encoded on an artificially-generated propagated signal, e.g., a machine-generated electrical, optical, or electromagnetic signal, that is generated to encode information for transmission to suitable receiver apparatus for execution by a data processing apparatus.

[0082] The term "data processing apparatus" refers to data processing hardware and encompasses all kinds of apparatus, devices, and machines for processing data, including by way of example a programmable processor, a computer, or multiple processors or computers. The apparatus can also be, or further include, special purpose logic circuitry, e.g., an FPGA (field programmable gate array) or an ASIC (application-specific integrated circuit). The apparatus can optionally include, in addition to hardware, code that creates an execution environment for computer programs, e.g., code that constitutes processor firmware, a protocol stack, a database management system, an operating system, or a combination of one or more of them.

[0083] A computer program, which may also be referred to or described as a program, software, a software application, an app, a module, a software module, a script, or code, can be written in any form of programming language, including compiled or interpreted languages, or declarative or procedural languages; and it can be deployed in any form, including as a stand-alone program or as a module, component, subroutine, or other unit suitable for use in a computing environment. A program may, but need not, correspond to a file in a file system. A program can be stored in a portion of a file that holds other programs or data, e.g., one or more scripts stored in a markup language document, in a single file dedicated to the program in question, or in multiple coordinated files, e.g., files that store one or more modules, sub-programs, or portions of code. A computer program can be deployed to be executed on one computer or on multiple computers that are located at one site or distributed across multiple sites and interconnected by a data communication network.

[0084] In this specification the term "engine" is used broadly to refer to a software-based system, subsystem, or process that is programmed to perform one or more specific functions. Generally, an engine will be implemented as one or more software modules or components, installed on one or more computers in one or more locations. In some cases, one or more computers will be dedicated to a particular engine; in other cases, multiple engines can be installed and running on the same computer or computers.

[0085] The processes and logic flows described in this specification can be performed by one or more programmable computers executing one or more computer programs to perform functions by operating on input data and generating output. The processes and logic flows can also be performed by special purpose logic circuitry, e.g., an FPGA or an ASIC, or by a combination of special purpose logic circuitry and one or more programmed computers.

[0086] Computers suitable for the execution of a computer program can be based on general or special purpose microprocessors or both, or any other kind of central processing unit.

Generally, a central processing unit will receive instructions and data from a read-only memory or a random access memory or both. The essential elements of a computer are a central processing unit for performing or executing instructions and one or more memory devices for storing instructions and data. The central processing unit and the memory can be supplemented by, or incorporated in, special purpose logic circuitry. Generally, a computer will also include, or be operatively coupled to receive data from or transfer data to, or both, one or more mass storage devices for storing data, e.g., magnetic, magneto-optical disks, or optical disks. However, a computer need not have such devices. Moreover, a computer can be embedded in another device, e.g., a mobile telephone, a personal digital assistant (PDA), a mobile audio or video player, a game console, a Global Positioning System (GPS) receiver, or a portable storage device, e.g., a universal serial bus (USB) flash drive, to name just a few.

[0087] Computer-readable media suitable for storing computer program instructions and data include all forms of non-volatile memory, media and memory devices, including by way of example semiconductor memory devices, e.g., EPROM, EEPROM, and flash memory devices; magnetic disks, e.g., internal hard disks or removable disks; magneto-optical disks; and CD-ROM and DVD-ROM disks.

[0088] To provide for interaction with a user, embodiments of the subject matter described in this specification can be implemented on a computer having a display device, e.g., a CRT (cathode ray tube) or LCD (liquid crystal display) monitor, for displaying information to the user and a keyboard and a pointing device, e.g., a mouse or a trackball, by which the user can provide input to the computer. Other kinds of devices can be used to provide for interaction with a user as well; for example, feedback provided to the user can be any form of sensory feedback, e.g., visual feedback, auditory feedback, or tactile feedback; and input from the user can be received in any form, including acoustic, speech, or tactile input. In addition, a computer can interact with a user by sending documents to and receiving documents from a device that is used by the user; for example, by sending web pages to a web browser on a user's device in response to requests received from the web browser. Also, a computer can interact with a user by sending text messages or other forms of message to a personal device, e.g., a smartphone that is running a messaging application, and receiving responsive messages from the user in return.

[0089] Data processing apparatus for implementing machine learning models can also include, for example, special-purpose hardware accelerator units for processing common and compute-intensive parts of machine learning training or production, i.e., inference, workloads.

[0090] Machine learning models can be implemented and deployed using a machine learning framework, e.g., a TensorFlow framework, or a Jax framework.

[0091] Embodiments of the subject matter described in this specification can be implemented in a computing system that includes a back-end component, e.g., as a data server, or that includes a middleware component, e.g., an application server, or that includes a front-end component, e.g., a client computer having a graphical user interface, a web browser, or an app through which a user can interact with an implementation of the subject matter described in this specification, or any combination of one or more such back-end, middleware, or front-end components. The components of the system can be interconnected by any form or medium of digital data communication, e.g., a communication network. Examples of communication networks include a local area network (LAN) and a wide area network (WAN), e.g., the Internet.

[0092] The computing system can include clients and servers. A client and server are generally remote from each other and typically interact through a communication network. The relationship of client and server arises by virtue of computer programs running on the respective computers and having a client-server relationship to each other. In some embodiments, a server transmits data, e.g., an HTML page, to a user device, e.g., for purposes of displaying data to and receiving user input from a user interacting with the device, which acts as a client. Data generated at the user device, e.g., a result of the user interaction, can be received at the server from the device.

[0093] While this specification contains many specific implementation details, these should not be construed as limitations on the scope of any invention or on the scope of what may be claimed, but rather as descriptions of features that may be specific to particular embodiments of particular inventions. Certain features that are described in this specification in the context of separate embodiments can also be implemented in combination in a single embodiment. Conversely, various features that are described in the context of a single embodiment can also be implemented in multiple embodiments separately or in any suitable subcombination. Moreover, although features may be described above as acting in certain combinations and even initially be claimed as such, one or more features from a claimed combination can in some cases be excised from the combination, and the claimed combination may be directed to a subcombination or variation of a subcombination.

[0094] Similarly, while operations are depicted in the drawings and recited in the claims in a particular order, this should not be understood as requiring that such operations be performed in the particular order shown or in sequential order, or that all illustrated operations be

performed, to achieve desirable results. In certain circumstances, multitasking and parallel processing may be advantageous. Moreover, the separation of various system modules and components in the embodiments described above should not be understood as requiring such separation in all embodiments, and it should be understood that the described program components and systems can generally be integrated together in a single software product or packaged into multiple software products.

[0095] Particular embodiments of the subject matter have been described. Other embodiments are within the scope of the following claims. For example, the actions recited in the claims can be performed in a different order and still achieve desirable results. As one example, the processes depicted in the accompanying figures do not necessarily require the particular order shown, or sequential order, to achieve desirable results. In some cases, multitasking and parallel processing may be advantageous.

[0096] Aspects of the present disclosure may be as set out in the following clauses:

Clause 1. A method performed by one or more computers, the method comprising:

receiving a network input;

processing the network input using a pre-trained neural network, the pre-trained neural network comprising a plurality of layers;

generating, from the outputs of one or more of the layers, features of the network input; and

generating a first out-of-distribution (OOD) score for the network input that represents the deviation of the network input from a distribution determined by a generative model configured to characterize a particular distribution of a first subset of training data.

Clause 2. The method of clause 1, wherein generating, from the outputs of one or more of the layers, features of the network input comprises:

selecting, as the features, a compressed representation of the values in the outputs of the one or more layers.

Clause 3. The method of clause 2, wherein the compressed representation has been identified by performing feature selection on respective outputs of the one or more layers for each of a plurality of training inputs in the first subset of training data.

Clause 4. The method of clause 3, wherein performing feature selection comprises performing principal component analysis (PCA) on the respective outputs of the one or more layers for each of a plurality of training inputs in the subset of training data.

Clause 5. The method of any preceding clause, wherein the distribution determined by the generative model comprises a mixture of one or more component distributions and wherein generating the first OOD score for the network input that represents the deviation of the network input from the distribution comprises calculating, for each component distribution, a deviation between the features and the component distribution.

Clause 6. The method of clause 5, wherein the generative model is a Gaussian Mixture Model (GMM) that determines a distribution that includes one or more component Gaussian distributions.

Clause 7. The method of clause 5 or 6, wherein the calculated deviation between the features and each component distribution is a Mahalanobis distance between the features and the component distribution.

Clause 8. The method of clause 5, clause 6, or clause 7, wherein generating the first OOD score further comprises selecting the minimum deviation calculated between the features and the component distributions.

Clause 9. The method of any preceding clause, further comprising:
generating a classification of the network input using a classifier neural network configured to perform classification into a respective one of a plurality of classes.

Clause 10. The method of clause 9, wherein the pre-trained neural network is the classifier neural network.

Clause 11. The method of clause 9 or clause 10, further comprising:
using at least the generated first OOD score to refine the classification of the network input generated by the classifier neural network.

Clause 12. The method of clause 11, wherein using at least the generated first OOD score to refine the classification comprises using at least the generated first OOD score and one or more subsequent OOD scores, each representing a deviation of the network input from a distribution of a corresponding subsequent subset of training data.

Clause 13. The method of clauses 9-12, wherein the classifier neural network is an image classification neural network.

Clause 14. The method of any preceding clause, wherein the network input is an image.

Clause 15. A system comprising one or more computers and one or more storage devices storing instructions that are operable, when executed by the one or more computers, to cause the one or more computers to perform the operations of the respective method of any preceding clause.

Clause 16. One or more computer-readable storage media encoded with instructions that, when executed by one or more computers, cause the one or more computers to perform the operations of the respective method of any one of clauses 1-14.

[0097] What is claimed is:

CLAIMS

1. A method performed by one or more computers, the method comprising:

receiving a network input;

processing the network input using a pre-trained neural network, the pre-trained neural network comprising a plurality of layers;

generating, from the outputs of one or more of the layers, features of the network input; and

generating a first out-of-distribution (OOD) score for the network input that represents the deviation of the network input from a distribution determined by a generative model configured to characterize a particular distribution of a first subset of training data.

2. The method of claim 1, wherein generating, from the outputs of one or more of the layers, features of the network input comprises:

selecting, as the features, a compressed representation of the values in the outputs of the one or more layers.

3. The method of claim 2, wherein the compressed representation has been identified by performing feature selection on respective outputs of the one or more layers for each of a plurality of training inputs in the first subset of training data.

4. The method of claim 3, wherein performing feature selection comprises performing principal component analysis (PCA) on the respective outputs of the one or more layers for each of a plurality of training inputs in the subset of training data.

5. The method of any preceding claim, wherein the distribution determined by the generative model comprises a mixture of one or more component distributions and wherein generating the first OOD score for the network input that represents the deviation of the network input from the distribution comprises calculating, for each component distribution, a deviation between the features and the component distribution.

6. The method of claim 5, wherein the generative model is a Gaussian Mixture Model (GMM) that determines a distribution that includes one or more component Gaussian distributions.

7. The method of claim 5 or 6, wherein the calculated deviation between the features and each component distribution is a Mahalanobis distance between the features and the component distribution.

8. The method of claim 5, claim 6, or claim 7, wherein generating the first OOD score further comprises selecting the minimum deviation calculated between the features and the component distributions.

9. The method of any preceding claim, further comprising:

generating a classification of the network input using a classifier neural network configured to perform classification into a respective one of a plurality of classes.

10. The method of claim 9, wherein the pre-trained neural network is the classifier neural network.

11. The method of claim 9 or claim 10, further comprising:

using at least the generated first OOD score to refine the classification of the network input generated by the classifier neural network.

12. The method of claim 11, wherein using at least the generated first OOD score to refine the classification comprises using at least the generated first OOD score and one or more subsequent OOD scores, each representing a deviation of the network input from a distribution of a corresponding subsequent subset of training data.

13. The method of claims 9-12, wherein the classifier neural network is an image classification neural network.

14. The method of any preceding claim, wherein the network input is an image.

15. A system comprising one or more computers and one or more storage devices storing instructions that are operable, when executed by the one or more computers, to cause the one

or more computers to perform the operations of the respective method of any preceding claim.

16. One or more computer-readable storage media encoded with instructions that, when executed by one or more computers, cause the one or more computers to perform the operations of the respective method of any preceding claim.

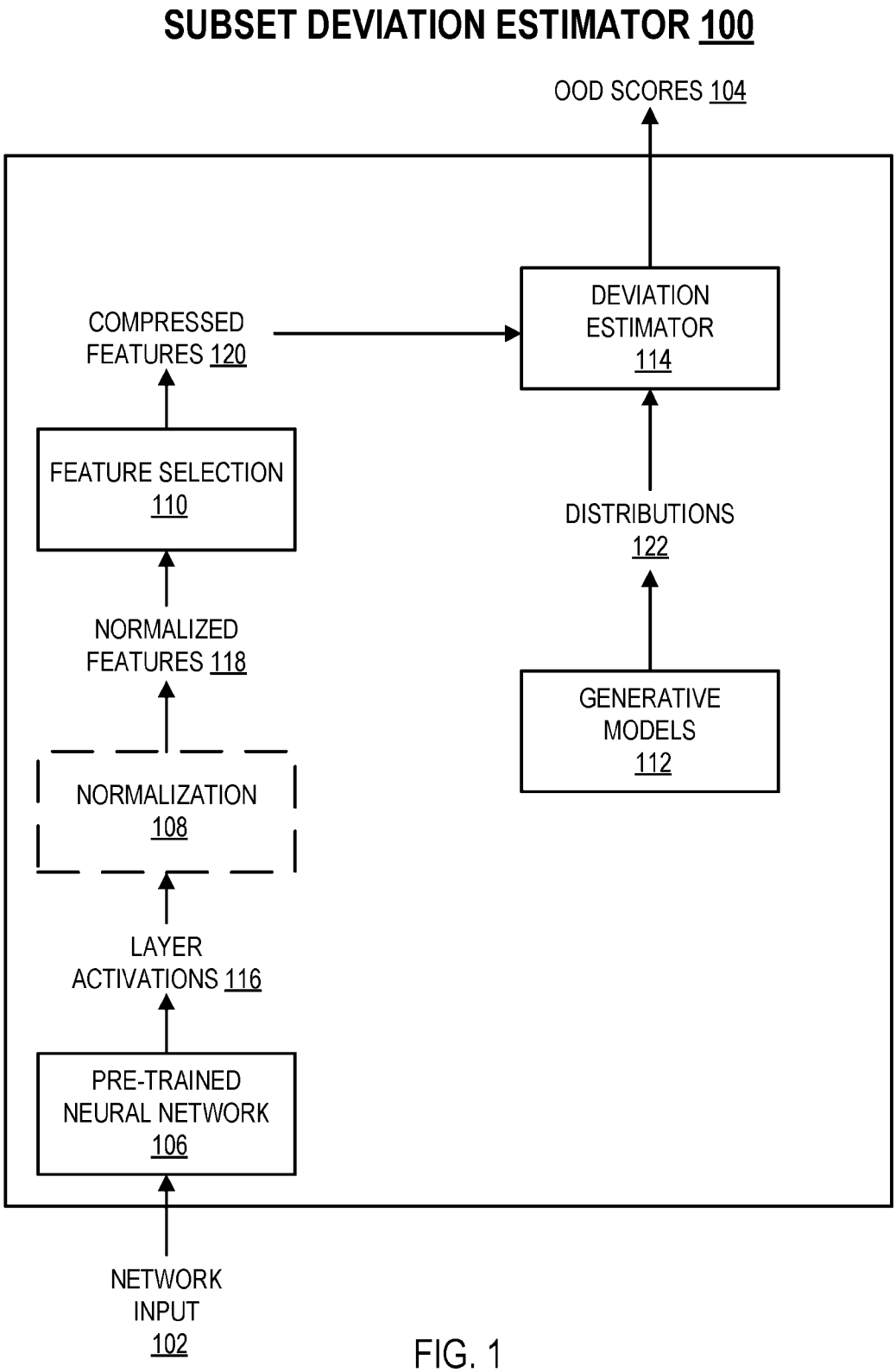

SUBSET DEVIATION ESTIMATOR 100
OOD SCORES 104
DEVIATION ESTIMATOR 114
COMPRESSED FEATURES 120
FEATURE SELECTION 110
DISTRIBUTIONS 122
NORMALIZED FEATURES 118
GENERATIVE MODELS 112
NORMALIZATION 108
LAYER ACTIVATIONS 116
PRE-TRAINED NEURAL NETWORK 106
NETWORK INPUT 102

FIG. 1

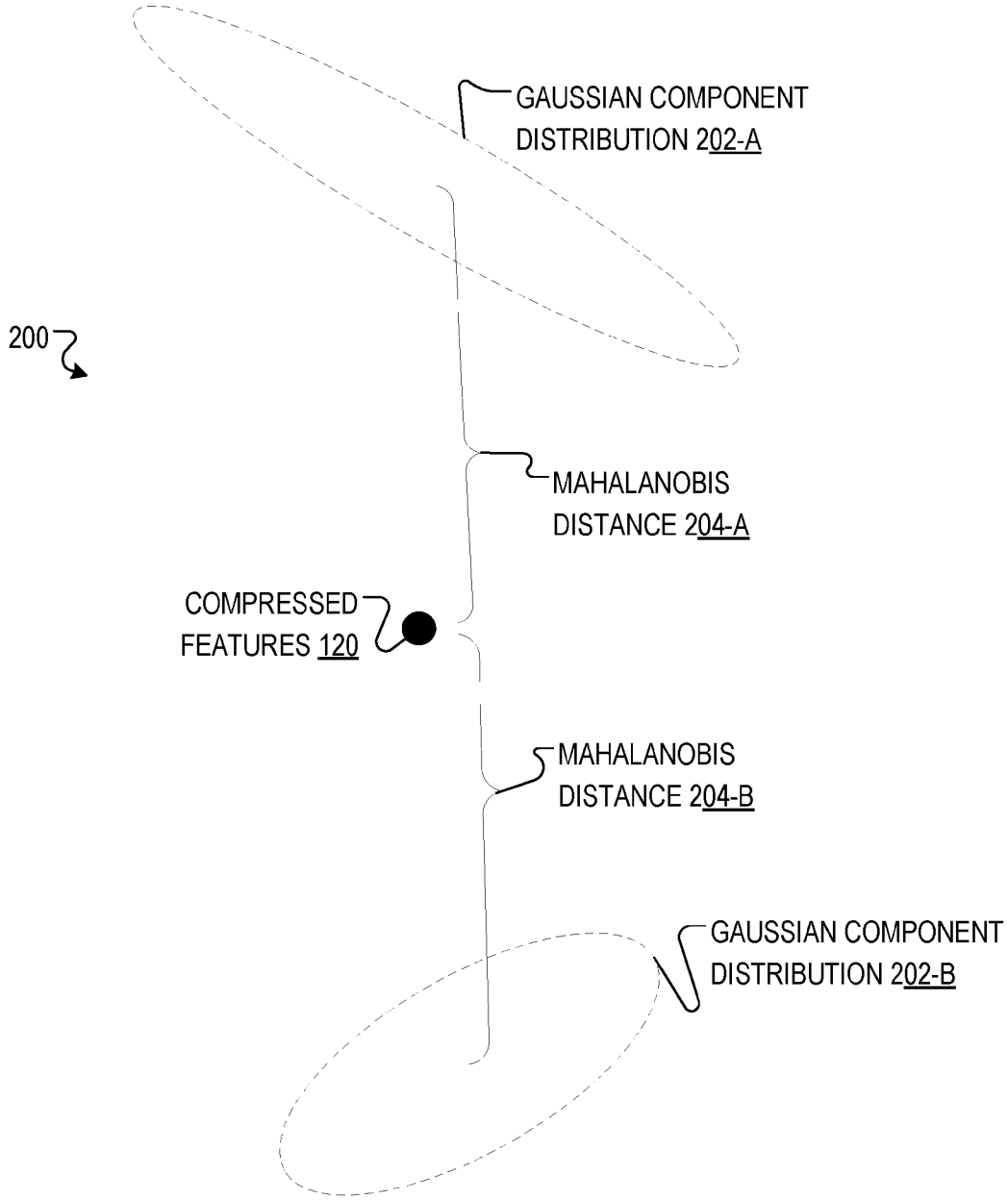

GAUSSIAN COMPONENT DISTRIBUTION 202-A
200
MAHALANOBIS DISTANCE 204-A
COMPRESSED FEATURES 120
MAHALANOBIS DISTANCE 204-B
GAUSSIAN COMPONENT DISTRIBUTION 202-B

FIG. 2

OUT-OF-DISTRIBUTION (OOD) CLASSIFICATION SYSTEM 300

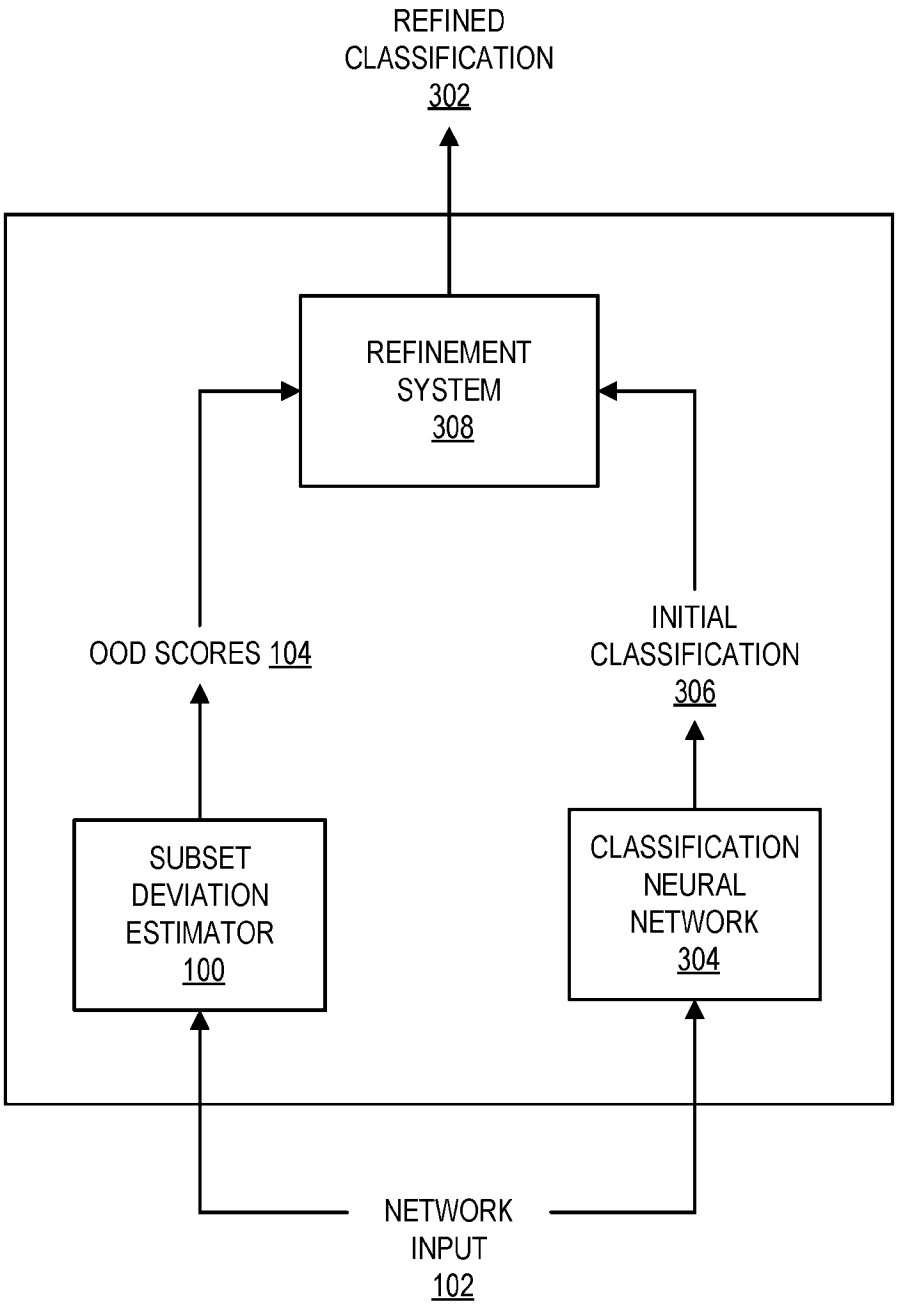


FIG. 3

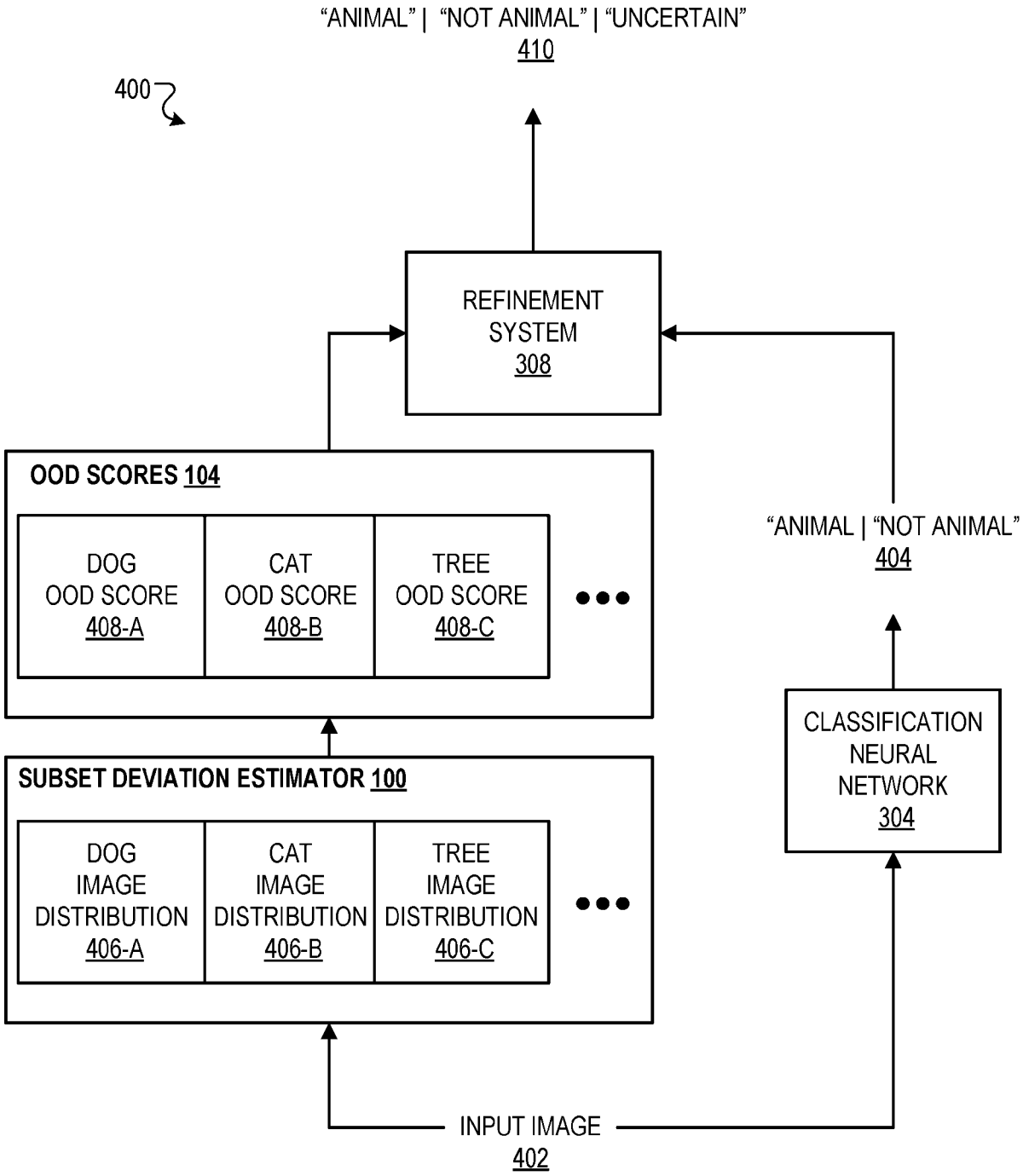

"ANIMAL" | "NOT ANIMAL" | "UNCERTAIN"
410
400
REFINEMENT SYSTEM 308
OOD SCORES 104
DOG OOD SCORE 408-A
CAT OOD SCORE 408-B
TREE OOD SCORE 408-C
"ANIMAL | "NOT ANIMAL"
404
CLASSIFICATION NEURAL NETWORK 304
SUBSET DEVIATION ESTIMATOR 100
DOG IMAGE DISTRIBUTION 406-A
CAT IMAGE DISTRIBUTION 406-B
TREE IMAGE DISTRIBUTION 406-C
INPUT IMAGE 402

FIG. 4

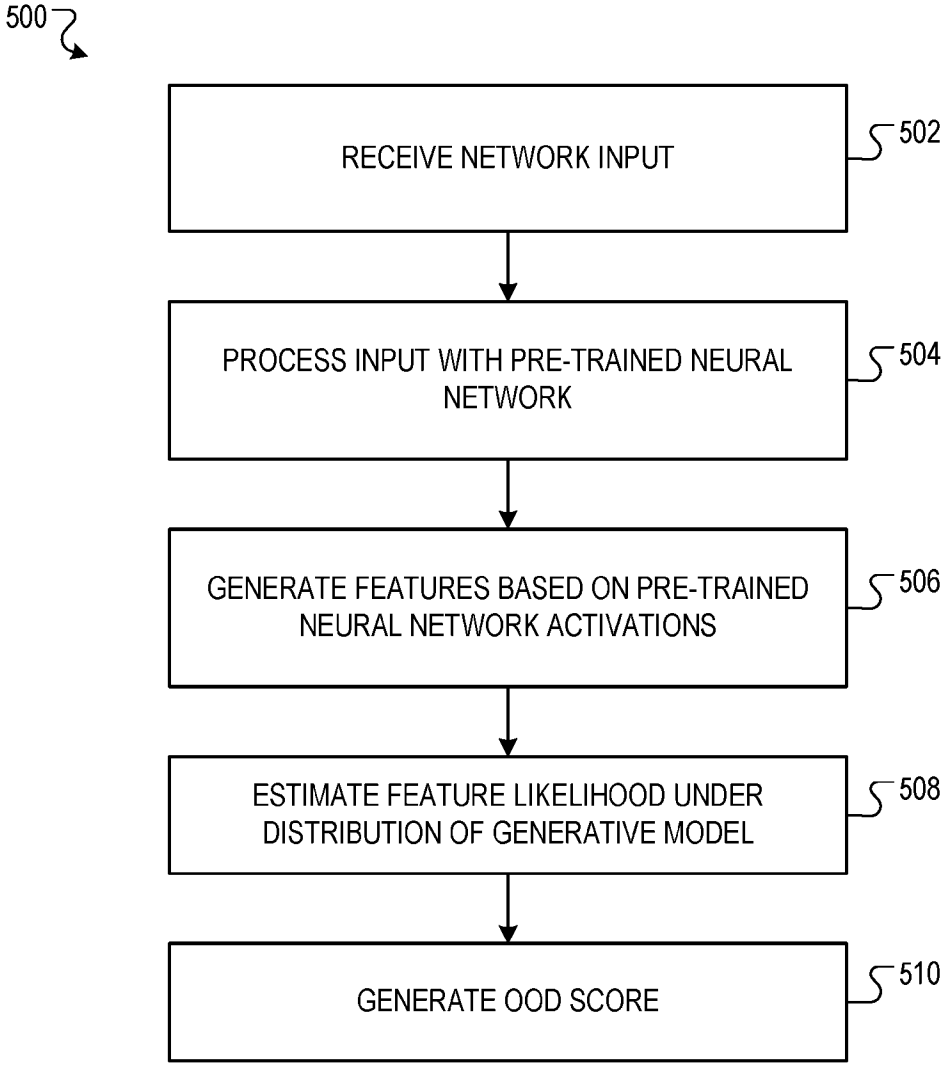

500
RECEIVE NETWORK INPUT
502
PROCESS INPUT WITH PRE-TRAINED NEURAL NETWORK
504
GENERATE FEATURES BASED ON PRE-TRAINED NEURAL NETWORK ACTIVATIONS
506
ESTIMATE FEATURE LIKELIHOOD UNDER DISTRIBUTION OF GENERATIVE MODEL
508
GENERATE OOD SCORE
510

FIG. 5

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2023/034282

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F18/2321 G06N3/045 G06N3/0475 G06N3/088 G06F18/23
G06F18/21
ADD.
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06N G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	JINGWEI XU ET AL: "Joint Distribution across Representation Space for Out-of-Distribution Detection", ARXIV.ORG, 30 March 2021 (2021-03-30), XP081914884, abstract page 1, left-hand column, paragraph 1 page 2, right-hand column, paragraph 6 page 3, left-hand column, paragraph 3; figure 1 page 4, left-hand column, paragraph 2 page 4, left-hand column, last paragraph - right-hand column, paragraph 1 page 5, left-hand column, paragraph 1 - right-hand column, paragraph 2 page 6, right-hand column, paragraph 2 - paragraph 4 page 6, right-hand column, paragraph 4 page 7; table 1 -/--	1-16

[X] Further documents are listed in the continuation of Box C. [ ] See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance;; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance;; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
19 April 2024	07/05/2024

Name and mailing address of the ISA/
European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040,
Fax: (+31-70) 340-3016

Authorized officer
Herri, Edmond

1

Form PCT/ISA/210 (second sheet) (April 2005)

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2023/034282

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	ADAMS STEPHEN ET AL: "A survey of feature selection methods for Gaussian mixture models and hidden Markov models", ARTIFICIAL INTELLIGENCE REVIEW, SPRINGER NETHERLANDS, NL, vol. 52, no. 3, 25 September 2017 (2017-09-25), pages 1739-1779, XP036884513, ISSN: 0269-2821, DOI: 10.1007/S10462-017-9581-3 [retrieved on 2017-09-25] abstract page 1746, paragraph 2	2-4
A	NDIOUR IBRAHIMA J ET AL: "Subspace Modeling for Fast Out-Of-Distribution and Anomaly Detection", 2022 IEEE INTERNATIONAL CONFERENCE ON IMAGE PROCESSING (ICIP), IEEE, 16 October 2022 (2022-10-16), pages 3041-3045, XP034292660, DOI: 10.1109/ICIP46576.2022.9897694 [retrieved on 2023-02-09] page 3042, left-hand column, paragraph 2 - page 3043, left-hand column, paragraph 2	1-16
A	LUKAS RUFF ET AL: "A Unifying Review of Deep and Shallow Anomaly Detection", ARXIV.ORG, 28 September 2020 (2020-09-28), XP081795097, page 3, left-hand column, paragraph 3 - right-hand column page 8, left-hand column, paragraph 4 - paragraph 5 page 13, right-hand column, paragraph 5 - page 14, left-hand column, paragraph 2 page 21, right-hand column, paragraph 6 - page 22, left-hand column, paragraph 1	1-16

1