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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
15 March 2018 (15.03.2018)

W!P O | PCT

(10) International Publication Number
WO 2018/048934 Al

(51) International Patent Classification:
G06N 3/04 *(2006.01)*

(21) International Application Number:
PCT/US20 17/050320

(22) International Filing Date:
06 September 2017 (06.09.2017)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
62/384,1 15 06 September 2016 (06.09.2016) US

(71) Applicant: **DEEPMIND TECHNOLOGIES LIMITED** [GB/GB]; 5 New Street Square, London EC4A 3TW (GB).

(72) Inventors: **VAN DEN OORD, Aaron Gerard Antonius;** 7 Pancras Square Kings Cross, London NIC 4AG (GB). **DIELEMAN, Sander Etienne Lea;** 7 Pancras Square, Kings Cross, London NIC 4AG (GB). **KALCHBRENNER, Nal Emmerich;** 7 Pancras Square, Kings Cross, London NIC 4AG (GB). **SIMONY AN, Karen;** 6 Pancras Square, London NIC 4AG (GB). **VINYALS, Oriol;** 6 Pancras Square, London NIC 4AG (GB).

(74) Agent: **PORTNOV, Michael** et al; FISH & RICHARDSON P.C., P.O. BOX 1022, 3300 RBC PLAZA, MINNEAPOLIS, Minnesota 55440-1022 (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, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, 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, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available):* ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, 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, 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).

(54) **Title:** GENERATING AUDIO USING NEURAL NETWORKS

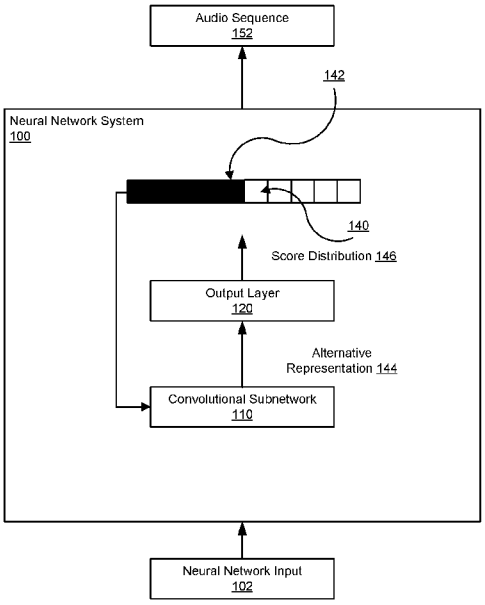

FIG. 1

(57) **Abstract:** Methods, systems, and apparatus, including computer programs encoded on computer storage media, for generating an output sequence of audio data that comprises a respective audio sample at each of a plurality of time steps. One of the methods includes, for each of the time steps: providing a current sequence of audio data as input to a convolutional subnetwork, wherein the current sequence comprises the respective audio sample at each time step that precedes the time step in the output sequence, and wherein the convolutional subnetwork is configured to process the current sequence of audio data to generate an alternative representation for the time step; and providing the alternative representation for the time step as input to an output layer, wherein the output layer is configured to: process the alternative representation to generate an output that defines a score distribution over a plurality of possible audio samples for the time step.

WO 2018/048934 A1

[Continued on nextpage]

Published:

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

— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))*

GENERATING AUDIO USING NEURAL NETWORKS

BACKGROUND

This specification relates to processing and generating audio using neural networks.

Neural networks are machine learning models that employ one or more layers of nonlinear units to predict an output for a received input. Some neural networks include one or more hidden layers in addition to an output layer. The output of each hidden layer is used as input to the next layer in the network, i.e., the next hidden layer or the output layer. Each layer of the network generates an output from a received input in accordance with current values of a respective set of parameters.

SUMMARY

This specification describes how a system implemented as computer programs on one or more computers in one or more locations can generate a sequence of audio data that includes a respective audio sample at each of multiple time steps. For example, the sequence of audio data can represent speech in a particular natural language or a piece of music.

In one innovative aspect a neural network system implemented by one or more computers is configured to generate an output sequence of audio data that comprises a respective audio sample at each of a plurality of time steps. The neural network system may comprise a convolutional subnetwork comprising one or more audio-processing convolutional neural network layers; and an output layer. The convolutional subnetwork may be configured to, for each of the plurality of time steps: receive a current sequence of audio data that comprises the respective audio sample at each time step that precedes the (current) time step in the output sequence. The convolutional subnetwork may further be configured to process the current sequence of audio data to generate an alternative representation for the time (current) step. This alternative representation may thus comprise a numeric representation, i.e., an ordered collection of numeric values, in which the current sequence of audio data has been encoded by the convolutional subnetwork, for example encoding features of the current sequence. The output layer may be configured to, for each of the plurality of time steps: receive the alternative representation for the time step, and process the alternative representation for the time step to generate an

output that defines a score distribution over a plurality of possible audio samples for the time step.

Some of the many advantages of such a system are described later. The system can use the score distribution to select a sample for the current time step, by sampling from the distribution. The output may, but need not necessarily, comprise one score for each possible audio sample value, for example 256 scores for 256 possible values. In can thus be useful to compress or compand the audio sample values, which may be amplitude values, to reduce the number of model outputs.

In some implementations the convolutional neural network layers are causal convolutional neural network layers, as described in more detail later. In particular, the audio-processing convolutional neural network layers may include one or more dilated causal convolutional neural network layers. Again as described in more detail later, a dilated convolutional neural network layer applies a convolution to non-adjacent values in a sequence, i.e., as defined by the outputs from a previous layer. This can increase the receptive field of the convolutional subnetwork by orders of magnitude whilst preserving the input (time) resolution and maintaining computational efficiency.

In some implementations the convolutional neural network layers include multiple stacked blocks of dilated convolutional neural network layers. Each block may comprise multiple dilated convolutional neural network layers with increasing dilation. For example the dilation may be increased by a factor n for each successive layer up to a limit within each block. This can further increase the receptive field size.

In some implementations one or more of the convolutional neural network layers may have gated activation units. For example a rectified linear or other unit following a convolution implemented by a layer may be replaced by a gated activation unit. In a gated activation unit the output may be a combination of two (causal) convolutions, a main convolution and a gate convolution. The convolutions may each be applied to some or all of the same outputs from the previous layer. The combination may involve a non-linear activation function applied to the gate convolution, for example an activation with a (0,1) range such as a sigmoid. This may then multiply a value from the main convolution; a non-linear activation function may, but need not be, applied to the main convolution. Such an approach may assist in capturing more complex structure within the data.

The alternative representation from the convolutional subnetwork at each time step may be conditioned on a neural network input, for example a latent representation of

a conditioning input. The conditioning input may be global (substantially time-independent) and/or local (time-dependent). The conditioning input may comprise, for example, text, image or video data, or audio data, for example an example of a particular speaker or language or music. The neural network input may comprise an embedding of the conditioning input. For example in a text-to-speech system a global conditioning input may comprise a speaker embedding and a local conditioning input may comprise linguistic features. The system may be configured to map the neural network input, or a conditioning input, from a lower sampling frequency to the audio sample generation frequency, for example by repeating the input or upsampling the input using a neural network. Thus the neural network input may comprise features of a text segment and the output sequence may represent a verbalization of the text segment; and/or the neural network input may comprise speaker or intonation pattern values; and/or the neural network input may include one or more of: speaker identity information, language identity information, and speaking style information. Alternatively the output sequence represents a piece of music.

The convolutional subnetwork may comprise residual connections, for example a connection from an input of a convolutional layer to a summer to sum this with an intermediate output of the layer. This effectively allows the network to be trained to skip or partially skip a layer, thus speeding up convergence and facilitating training of deeper models. The convolutional subnetwork may additionally or alternatively comprise skip connections, for example directly from each of one or more intermediate layers of the convolutional subnetwork to one or more operations that directly generate the alternative representation that is provided to the output layer.

In some implementations processing the current sequence of audio data using the convolutional subnetwork, to generate an alternative representation for the time step, re-uses values computed for previous time steps. The re-used values may comprise values derived from application of a convolutional filter to the audio sample data or data derived therefrom. The re-used values may be stored at one time step and retrieved at a later time step when the same filter is applied to the same (or some of the same) audio sample data or data derived therefrom. This can make the system computationally more efficient and hence faster, because there is no need to re-compute the stored values.

Particular embodiments of the subject matter described in this specification can be implemented so as to realize one or more of the following advantages. The neural network system can generate on the order of tens of thousands of audio samples per

second, providing a greater level of granularity than other neural network-based audio generation systems. The neural network system can achieve results that significantly outperform the state of the art on audio generation tasks, e.g., by generating speech from text that is of higher quality than state of the art techniques. A single trained neural network system can be used to generate different voices by conditioning on the speaker identity. By using convolutional neural network layers, e.g., causal convolutional layers, instead of recurrent neural network layers, e.g., instead of long short-term memory (LSTM) layers, the neural network system can achieve these advantageous results while not needing as many computational resources to train as other systems that do include recurrent neural network layers, resulting in a reduced training time. By employing convolutional layers rather than recurrent layers, the computation of the neural network system can be more easily batched and more easily parallelized, e.g.., because the layers of the network do not have to be unrolled for each time step, allowing the computation of the system to be performed more efficiently. Additionally, by employing dilated causal convolutional layers, the receptive field of the convolutional subnetwork and, therefore, the quality of the audio generated by the system, can be improved without greatly increasing the computational cost of generating the audio.

The details of one or more embodiments of the subject matter described in 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

FIG. 1 shows an example neural network system.

FIG. 2 shows a visualization of an example block of dilated causal convolutional layers.

FIG. 3 shows an example architecture for the convolutional subnetwork.

FIG. 4 is a flow diagram of an example process for generating an audio sample at a given time step in an audio sequence.

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

DETAILED DESCRIPTION

FIG. 1 shows an example neural network system 100. The neural network 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 can be implemented.

The neural network system 100 generates sequences of audio data that each include a respective audio sample at each of multiple time steps, e.g., an output sequence of audio data 152.

Generally, each time step in a given audio sequence corresponds to a respective time in an audio waveform and the audio sample at the time step characterizes the waveform at the corresponding time. In some implementations, the audio sample at each time step in the sequence is the amplitude of the audio waveform at the corresponding time, i.e., the sequence generated by the neural network system 100 is a raw audio waveform. In some other implementations, the audio sample at each time step in the sequence is a compressed or companded representation of the waveform at the corresponding time. For example, the audio sample can be a μ-law transformed representation of the waveform.

More specifically, the neural network system 100 generates audio sequences autoregressively. That is, for each particular time step in an output audio sequence, the neural network system 100 generates the audio sample at the time step conditioned on the audio samples that have already been generated as of the particular time step, i.e., on audio samples at time steps that are earlier than the particular time step in the audio sequence.

The neural network system 100 includes a convolutional subnetwork 110 and an output layer 120.

At each time step during the generation of an audio sequence, the convolutional subnetwork 110 is configured to receive the current audio sequence, i.e., the audio sequence that has already been generated as of the time step, and to process the current audio sequence to generate an alternative representation for the time step. For example, when generating an audio sample 140 in the audio sequence 152, the convolutional subnetwork 110 can receive a current audio sequence 142 that includes the audio samples that precede the audio sample 140 in the audio sequence 152 and process the current audio sequence 142 to generate an alternative representation 144.

The output layer 120 is configured to, at each of the time steps, receive the alternative representation at the time step and generate a score distribution over possible audio samples for the time step. The score distribution includes a respective score for each of multiple possible audio samples. In some implementations, the output layer 120 is a softmax output layer. For example, the output layer 120 can receive the alternative representation 144 and process the alternative representation 144 to generate a score distribution 146.

In particular, when the neural network system 100 is configured to generate raw audio data, the score distribution includes a respective score for each of multiple possible amplitude values. When the neural network system 100 is configured to generate compressed or companded values, the score distribution includes a respective score for each of multiple possible compressed or companded values.

Once the output layer 146 has generated the score distribution for a given time step, the neural network system 100 can select an audio sample to be included in the output sequence at the given time step from the multiple possible audio samples in accordance with the score distribution for the given time step. For example, the neural network system 100 can select an audio sample by sampling from the score distribution, i.e., sampling from the possible audio samples in accordance with the scores in the score distribution so that each audio sample is selected with a likelihood that corresponds to the score for the audio sample, or can select the possible audio sample having the highest score according to the score distribution.

The convolutional subnetwork 110 generally includes multiple audio-processing convolutional neural network layers. More specifically, the audio-processing convolutional neural network layers include multiple causal convolutional layers.

A causal convolutional layer is a convolutional layer that operates on an input sequence that has a respective input at each of multiple time steps by, for each time step, generating an output that depends only on the inputs at the time step and at the time steps before the time step in the input sequence, i.e., and not on any inputs at any time steps after the time step in the input sequence. In some cases, the causal convolutional layers are implemented by applying a normal convolution and then shifting each output of the normal convolution by a few time steps, i.e., shifting each output forward by (filter length - 1) time steps, prior to applying the activation function for the convolutional layer, where "filter length" is the length of the filter of the convolution that is being applied.

To increase the receptive field of the audio-processing convolutional layers without requiring an excessive number of layers or filters of excessive length, some or all of the audio-processing convolutional layers can be dilated causal convolutional layers. A dilated convolution is a convolution where the filter is applied over an area larger than its length by skipping input values with a certain step that is defined by the dilation value for the dilated convolution. By incorporating dilated causal convolutions, the audio-processing neural network layers effectively operate on their inputs with a coarser scale than with a normal convolution.

In some implementations, the audio-processing neural network layers include a stack of multiple blocks of dilated causal convolutional layers. Each block in the stack can include multiple dilated convolutional neural network layers with increasing dilation. For example, within a block, the dilation can double for each layer starting from an initial dilation, and then return to the initial dilation for the first layer in the next block. As an illustrative example, the dilations of the dilated convolutional layers in a block can be, in order: $1, 2, 4, \ldots, 512$. A simplified example of a block of dilated causal convolutional layers is described below with reference to FIG. 2.

In some implementations, the convolutional subnetwork includes residual connections, skip connections, or both. An example architecture of the convolutional subnetwork that includes both residual connections and skip connections is described below with reference to FIG. 3.

In some implementations, the neural network system 100 generates audio sequences conditioned on a neural network input. For example, the neural network system 100 can generate the audio sequence 152 conditioned on a neural network input 102.

In some cases, the neural network input includes one or more local features, i.e., one or more features that are different for different time steps in the output sequence. For example, the neural network system 100 can obtain as input linguistic features of a text segment and can generate an audio sequence that represents a verbalization of the text segment, i.e., the neural network system 100 can function as part of a text-to-speech system that converts written text to spoken speech and also includes a component that verbalizes the audio sequence generated by the neural network system 100.

In some other cases, the neural network input includes one or more global features, i.e., one or more features that are the same throughout the entire output sequence. As an example, the neural network system 100 can generate speech conditioned

on an identity of the speaker, i.e., so that the speech is generated to sound like the voice of the speaker. In this example, the neural network system 100 can obtain a vector encoding the identity of the speaker, e.g., a one-hot encoded vector identifying the speaker, and condition the generated speech on the obtained vector.

Generally, the audio sequences are conditioned on the neural network input by conditioning the activation function of some or all of the convolutional layers in the convolutional subnetwork. That is, the output of the activation function and, accordingly, the output of the convolutional layer, is dependent not only on the output of the convolution performed by the layer but also on the neural network input.

Conditioning an activation function of a convolutional layer on the neural network input will be described in more detail below with reference to FIG. 3.

FIG. 2 shows a visualization 200 of an example block of dilated causal convolutional layers. In particular, the example block includes a dilated causal convolutional layer 204 with dilation one, a dilated causal convolutional layer 206 with dilation two, a dilated causal convolutional layer 208 with dilation four, and a dilated causal convolutional layer 210 with dilation eight.

In the visualization 200, the block of dilated causal convolutional layers are operating on a current input sequence 202 to generate an output sequence. In particular, the visualization 200 visualizes using bold arrows how the block generates the output 212 that is the output at the time step that is currently the last time step in the current input sequence 202 and the output sequence.

As can be seen from the visualization 200, because each layer in the block is a causal convolutional layer, the output 212 depends only on outputs that are at the last current time step or time steps before the last current time step in the various sequences operated on by the layers in the block.

Additionally, as can be seen from the visualization 200, the layers in the block are arranged in order of increasing dilation, with the first layer in the block, i.e., dilated causal convolutional layer 204, having dilation one and the last layer in the block, i.e., dilated causal convolutional layer 204, having dilation eight. In particular, as is shown by the bold arrows in the visualization 200, because the dilated causal convolutional layer 204 has dilation one, the filter of the layer 204 is applied to adjacent inputs in the current input sequence 202. Because the dilated causal convolutional layer 206 has dilation two, the filter of the layer 206 is applied to outputs that are separated by one output in the output sequence generated by the layer 204. Because the dilated causal convolutional

layer 208 has dilation four, the filter of the layer 208 is applied to outputs that are separated by three outputs in the output sequence generated by the layer 206. Because the dilated causal convolutional layer 210 has dilation eight, the filter of the layer 210 is applied to outputs that are separated by seven outputs in the output sequence generated by the layer 208.

FIG. 3 shows an example architecture 300 for the convolutional subnetwork 110 of FIG. 1. As described above, in the example architecture 300, the dilated causal convolutional layers that are in the convolutional subnetwork have residual connections and skip connections.

In particular, in the architecture 300, the convolutional subnetwork 110 includes a causal convolutional layer 302 that processes the current output sequence 142, i.e., by applying a causal convolution to the current output sequence 142.

The convolutional subnetwork 110 then processes the output of the causal convolutional layer 302 through a stack of dilated causal convolutional layers.

Each dilated causal convolutional layer 304 in the stack applies a dilated causal convolution 308 to the input 306 to the dilated causal convolutional layer 304. As described above, in some implementations, the dilated causal convolutional layers in the stack are arranged in blocks, with the dilation of the dilated causal convolutions applied by each layer increasing within a given block and then restarting at the initial value for the first layer in the next block.

In some implementations, the dilated causal convolutional layers in the stack have a gated activation function in which the output of an element-wise non-linearity, i.e., of a conventional activation function, is element-wise multiplied by a gate vector. In some of these implementations, the dilated causal convolution 308 includes two dilated causal convolutions on the layer input 302 - a first dilated causal convolution between a main filter for the layer 304 and the layer input 306 and another dilated causal convolution between a gate filter for the layer 304 and the layer input 306. In others of these implementations, dilated causal convolution 308 is a single dilated causal convolution and half of the output of the single convolution is provided as the output of the dilated causal convolution between the main filter for the layer 304 and the layer input 306 and the other half of the output of the single convolution is provided as the output of the dilated causal convolution between the gate filter for the layer 304 and the layer input 306.

The dilated causal convolutional layer 304 then determines the output of the activation function of the layer 304 using the outputs of the dilated causal convolution.

In particular, when the activation function is a gated activation function and the output sequence being generated is not conditioned on a neural network input, the layer 304 applies an element-wise non-linear function 310 which, in the example of FIG. 3 is the *tank* function, to the output of the dilated convolution with the main filter and applies an element-wise gating function which, in the example of FIG. 3, is the *sigmoid* function, to the output of the dilated convolution with the gate filter. The layer 304 then performs an element-wise multiplication 314 between the output of the non-linear function 310 and the output of the gating function 312 to generate the activation function output.

More specifically, when the element-wise non-linearity is *tank* and the element-wise gating function is the *sigmoid* function, the output of the activation function *z* for a layer *k* satisfies:

$$\mathbf{z} = \tanh(W_{f,k} * \chi) \odot \sigma(¾ * \mathbf{x}),$$

where $W_{f,k}$ is the main filter for the layer *k*, **x** is the layer input, * denotes a causal dilated convolution, $\odot$ denotes element-wise multiplication, and $W_{g,k}$ is the gate filter for the layer *k*.

When the output sequence being generated is conditioned on a neural network input, the layer 304 also conditions the output of the activation function on the neural network input. In particular, the non-linear function and the gating function each take as input a combination of the corresponding dilated convolution output and an input generated from the neural network input.

More specifically, when the neural network input includes global features and is therefore the same for all of the time steps in the sequence, the element-wise non-linearity is *tanh* and the element-wise gating function is the *sigmoid* function, the output of the activation function *z* for the layer *k* satisfies:

$$\mathbf{z} = \tanh(W_{f,k} * \mathbf{x} + V_{f,k}^T \mathbf{h}) \odot \sigma(W_{g,k} * \mathbf{x} + ¾ \mathbf{h}),$$

where $V_{f,k}^T$ is a main leamable linear projection (of **h** to the main component of the activation function) for the layer *k*, **h** is the neural network input, and $V_{f,k}^T$ is a gate leamable linear projection (of **h** to the gate component of the activation function) for the layer *k*.

Alternatively, when the neural network input includes local features, i.e., features that change from time step to time step, the system 100 obtains a sequence y that includes a set of features for each time step in the output sequence. The output of the activation function z for the layer *k* then satisfies:

$$\mathbf{z} = \tanh(W_{f,k} * \mathbf{x} + V_{f,k} * y) \odot \sigma(W_{g,k} * \mathbf{x} + V_{g,k} * \mathbf{y}),$$

where $V_{f,k} * y$ and $V_{g,k} * y$ are respective 1 x 1 convolutions. In some implementations, the system 100 directly receives the sequence **y** as the neural network input, i.e., directly receives a sequence that has the same resolution as the output sequence. In other implementations, the system 100 receives a sequence having a lower resolution, i.e., with a lower sampling frequency, than the output sequence. In these cases, the system can generate the sequence **y** by processing the lower resolution sequence using a transposed (learned upsampling) convolutional network to generate the sequence **y** or can repeat values from the lower resolution sequence across time to generate the sequence **y**.

As an example, when the local features are linguistic features for use in text to speech generation, the linguistic features can include some or all of phone, syllable, word, phrase, and utterance-level features of the text. Example sets of linguistic features that can be used are described in Zen, Heiga. *An example of context-dependent label format for HMM-based speech synthesis in English,* 2006. URL http://hts.sp.nitech.ac.jp/7Download and Zen, Heiga, Senior, Andrew, and Schuster, Mike. *Statistical parametric speech synthesis using deep neural networks.* In Proc. ICASSP, pp. 7962-7966, 2013.

Because the architecture 300 includes skip connections and residual connections for the dilated causal convolutional layers, the layer 304 then performs a 1 x 1 convolution 316 on the activation function output.

The layer 304 provides the output of the 1 x 1 convolution as the skip output 318 of the layer and adds the residual, i.e., the layer input 306, and the output of the 1 x 1 convolution to generate the final output 320 of the layer 304. The convolutional subnetwork 110 then provides the final output 320 as the layer input to the next dilated convolutional layer in the stack.

In some implementations, the layer 304 performs two 1 x 1 convolutions on the activation function output, one with a residual filter and the other with a skip filter. In these implementations, the layer 304 provides the output of the convolution with the skip filter as the skip output 318 of the layer and adds the residual and the output of the 1 x 1 convolution with the residual filter to generate the final output 320 of the layer 304.

The convolutional subnetwork 110 then provides the final output 320 as the layer input to the next dilated convolutional layer in the stack. For the last layer in the stack, because there is no next layer, the convolutional subnetwork 110 can either discard the

final output 320 generated by the last layer or can refrain from computing a final output, i.e., can refrain from performing the 1 x 1 convolution and the residual sum for the last layer in the stack.

Once the processing of all of the layers 304 in the stack of dilated convolutional layers has been completed, the convolutional subnetwork 110 sums 322 the skip outputs generated by the layers 304. The convolutional subnetwork 110 can then apply one or more non-linear functions, one or more 1 x 1 convolutions, or both to the sum 322 to generate the alternative representation 144. In particular, in the example of FIG. 3, the convolutional subnetwork 110 applies an element-wise non-linearity 324, e.g., a ReLU, followed by a lxl convolution 326, followed by another element-wise non-linearity 328, and followed by a final lxl convolution 330, to generate the alternative representation 144.

As described above, the output layer 120 then processes the alternative representation 144 to generate the score distribution 146.

FIG. 4 is a flow diagram of an example process 400 for generating an audio sample at a given time step in an audio sequence. For convenience, the process 400 will be described as being performed by a system of one or more computers located in one or more locations. For example, a neural network system, e.g., the neural network system 100 of FIG. 1, appropriately programmed, can perform the process 400.

The system provides a current audio sequence as input to the convolutional subnetwork (step 402). The current audio sequence is the audio sequence that has already been generated as of the given time step, i.e., a sequence that includes the output audio samples at time steps before the given time step. As described above, the convolutional subnetwork includes audio-processing convolutional neural network layers, e.g., dilated causal convolutional layers, and is configured to process the current sequence of audio data to generate an alternative representation for the given time step.

The system provides the alternative representation as input to an output layer, e.g., a softmax output layer (step 404). The output layer is configured to process the alternative representation to generate a score distribution over possible audio samples for the time step.

The system selects an audio sample for inclusion in the audio sequence at the given time step in accordance with the score distribution (step 406). For example, the system can sample a possible audio sample in accordance with the score distribution.

The system may be trained on raw or compressed and/or companded audio data, for example waveforms of human speakers, music and so forth. Optionally conditioning data may be included, for example text-to-speech data, which may be represented as linguistic features derived from text paired with audio data for a verbalization of the text. At training time, i.e., during the training of the convolutional subnetwork and the output layer to determine trained values of the filters of the convolutional layers and any other parameters of the system, the system can generate the conditional predictions for all time steps in parallel, i.e., instead of autoregressively, because all time steps of the ground truth output that should be generated by the system are known. Generally, the system can perform the training to determine the trained values of the parameters using conventional supervised learning techniques, e.g., a stochastic gradient descent with backpropagation based technique. As described above, because of this parallelization and the use of causal convolutional layers, the system does not need as many computational resources to train as other systems, e.g., those that include recurrent neural network layers, resulting in a reduced training time.

Additionally, because the system generates output sequences auto-regressively, in some implementations, the convolutional subnetwork reuses values computed for previous time steps when computing the alternative representation for the given time step. In particular, because the same audio samples are provided as input to the subnetwork more than once, some of the computation performed by the convolutional subnetwork will be the same at multiple different time steps. In these implementations, rather than re-compute these computations each time step, the convolutional subnetwork can store the output values of the computation the first time that the computation is performed and then re-use the stored output values at subsequent time steps. As a simple example, the first convolutional layer in the convolutional subnetwork will apply the same filter or filters multiple times to the same audio sample values during the generation of an audio sequence. Rather than re-compute the output of these filter applications at each time step, the system can re-use outputs computed at previous time steps.

In some implementations, as another way to increase the receptive field, one stack of dilated causal convolutional layers with a very large (long) receptive field, but preferably fewer units per layer, may be employed to condition another (larger) stack with a smaller receptive field. The larger stack may process a shorter part of the audio signal, for example cropped at the end.

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.

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.

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.

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.

In this specification, the term "database" is used broadly to refer to any collection of data: the data does not need to be structured in any particular way, or structured at all, and it can be stored on storage devices in one or more locations. Thus, for example, the index database can include multiple collections of data, each of which may be organized and accessed differently.

Similarly, 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.

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.

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.

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.

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.

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.

Machine learning models can be implemented and deployed using a machine learning framework, .e.g., a TensorFlow framework, a Microsoft Cognitive Toolkit framework, an Apache Singa framework, or an Apache MXNet framework.

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.

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.

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.

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.

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.

WHAT IS CLAIMED IS:

1. A neural network system implemented by one or more computers,

wherein the neural network system is configured to generate an output sequence of audio data that comprises a respective audio sample at each of a plurality of time steps, and

wherein the neural network system comprises:

a convolutional subnetwork comprising one or more audio-processing convolutional neural network layers, wherein the convolutional subnetwork is configured to, for each of the plurality of time steps:

receive a current sequence of audio data that comprises the respective audio sample at each time step that precedes the time step in the output sequence, and

process the current sequence of audio data to generate an alternative representation for the time step; and

an output layer, wherein the output layer is configured to, for each of the plurality of time steps:

receive the alternative representation for the time step, and

process the alternative representation for the time step to generate an output that defines a score distribution over a plurality of possible audio samples for the time step.

2. The neural network system of claim 1, wherein the neural network system further comprises:

a subsystem configured to, for each of the plurality of time steps:

select an audio sample at the time step in the output sequence in accordance with the score distribution for the time step.

3. The neural network system of claim 2, wherein selecting the audio value comprises:

sampling from the score distribution.

4. The neural network system of claim 2, wherein selecting the audio value comprises:

selecting an audio sample having a highest score according to the score distribution.

5. The neural network system of any one of claims 1-4, wherein each of the plurality of time steps corresponds to a respective time in an audio waveform, and wherein the respective audio sample at each of the plurality of time steps is an amplitude value of the audio waveform at the corresponding time.

6. The neural network system of any one of claims 1-4, wherein each of the plurality of time steps corresponds to a respective time in an audio waveform, and wherein the respective audio sample at each of the plurality of time steps is a compressed or a companded representation of the audio waveform at the corresponding time.

7. The neural network system of any one of claims 1-6, wherein the audio-processing convolutional neural network layers are causal convolutional neural network layers.

8. The neural network system of any one of claims 1-7, wherein the audio-processing convolutional neural network layers include one or more dilated convolutional neural network layers.

9. The neural network system of claim 8, wherein the audio-processing convolutional neural network layers include multiple blocks of dilated convolutional neural network layers, wherein each block comprises multiple dilated convolutional neural network layers with increasing dilation.

10. The neural network system of any one of claims 1-9, wherein one or more of the audio-processing convolutional neural network layers have gated activation units.

11. The neural network system of any one of claims 1-10, wherein, at each of the plurality of time steps, the alternative representation is conditioned on a neural network input.

12. The neural network system of claim 11, wherein the neural network input comprises features of a text segment, and wherein the output sequence represents a verbalization of the text segment.

13. The neural network system of claim 12, wherein the neural network input further comprises intonation pattern values.

14. The neural network system of any one of claims 11-13, wherein the neural network input comprises one or more of: speaker identity information, language identity information, and speaking style information.

15. The neural network system of any one of claims 1-14, wherein the output sequence represents a piece of music.

16. The neural network system of any one of claims 1-15, wherein the convolutional subnetwork comprises residual connections.

17. The neural network system of any one of claims 1-16, wherein the convolutional subnetwork comprises skip connections.

18. The neural network system of any one of claims 1-17, wherein processing the current sequence of audio data to generate an altemative representation for the time step comprises reusing values computed for previous time steps.

19. One or more computer storage media encoded with instructions that when executed by one or more computers cause the one or more computers to implement the respective neural network system of any one of claims 1-18.

20. A method of generating an output sequence of audio data that comprises a respective audio sample at each of a plurality of time steps,

wherein the method comprises, for each of the plurality of time steps:

providing a current sequence of audio data as input to a convolutional subnetwork comprising one or more audio-processing convolutional neural network layers,

wherein the current sequence comprises the respective audio sample at each time step that precedes the time step in the output sequence, and

wherein the convolutional subnetwork is configured to, for each of the plurality of time steps:

receive the current sequence of audio data, and

process the current sequence of audio data to generate an alternative representation for the time step; and

providing the alternative representation for the time step as input to an output layer, wherein the output layer is configured to, for each of the plurality of time steps:

receive the alternative representation for the time step, and

process the alternative representation for the time step to generate an output that defines a score distribution over a plurality of possible audio samples for the time step.

21. The method of claim 20, further comprising:

for each of the plurality of time steps:

selecting an audio sample at the time step in the output sequence in accordance with the score distribution for the time step.

22. The method of claim 21, wherein selecting the audio value comprises:

sampling from the score distribution.

23. The method of claim 21, wherein selecting the audio value comprises:

selecting an audio sample having a highest score according to the score distribution.

24. The method of any one of claims 20-23, wherein each of the plurality of time steps corresponds to a respective time in an audio waveform, and wherein the respective audio sample at each of the plurality of time steps is an amplitude value of the audio waveform at the corresponding time.

25. The method of any one of claims 20-23, wherein each of the plurality of time steps corresponds to a respective time in an audio waveform, and wherein the respective

audio sample at each of the plurality of time steps is a compressed or a companded representation of the audio waveform at the corresponding time.

26. The method of any one of claims 20-25, wherein the audio-processing convolutional neural network layers are causal convolutional neural network layers.

27. The method of any one of claims 20-26, wherein the audio-processing convolutional neural network layers include one or more dilated convolutional neural network layers.

28. The method of claim 27, wherein the audio-processing convolutional neural network layers include multiple blocks of dilated convolutional neural network layers, wherein each block comprises multiple dilated convolutional neural network layers with increasing dilation.

29. The method of any one of claims 20-28, wherein one or more of the audio-processing convolutional neural network layers have gated activation units.

30. The method of any one of claims 20-29, wherein, at each of the plurality of time steps, the alternative representation is conditioned on a neural network input.

31. The method of claim 30, wherein the neural network input comprises features of a text segment, and wherein the output sequence represents a verbalization of the text segment.

32. The method of claim 31, wherein the neural network input further comprises intonation partem values.

33. The method of any one of claims 30-32, wherein the neural network input comprises one or more of: speaker identity information, language identity information, and speaking style information.

34. The method of any one of claims 20-33, wherein the output sequence represents a piece of music.

35. The method of any one of claims 20-34, wherein the convolutional subnetwork comprises residual connections.

36. The method of any one of claims 20-35, wherein the convolutional subnetwork comprises skip connections.

37. The method of any one of claims 20-36, wherein processing the current sequence of audio data to generate an alternative representation for the time step comprises reusing values computed for previous time steps.

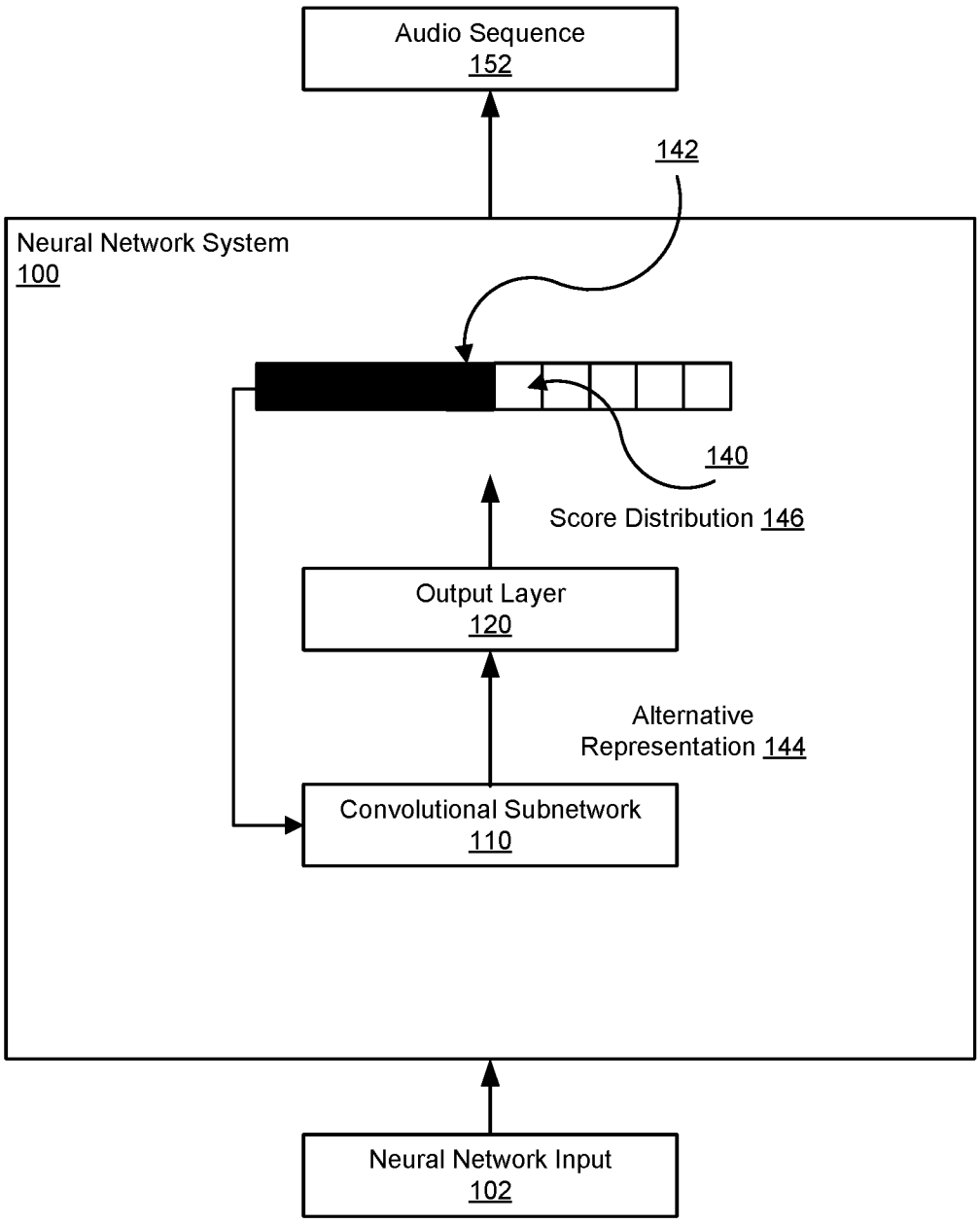
Audio Sequence
152
142
Neural Network System
100
140
Score Distribution 146
Output Layer
120
Alternative
Representation 144
Convolutional Subnetwork
110
Neural Network Input
102

FIG. 1

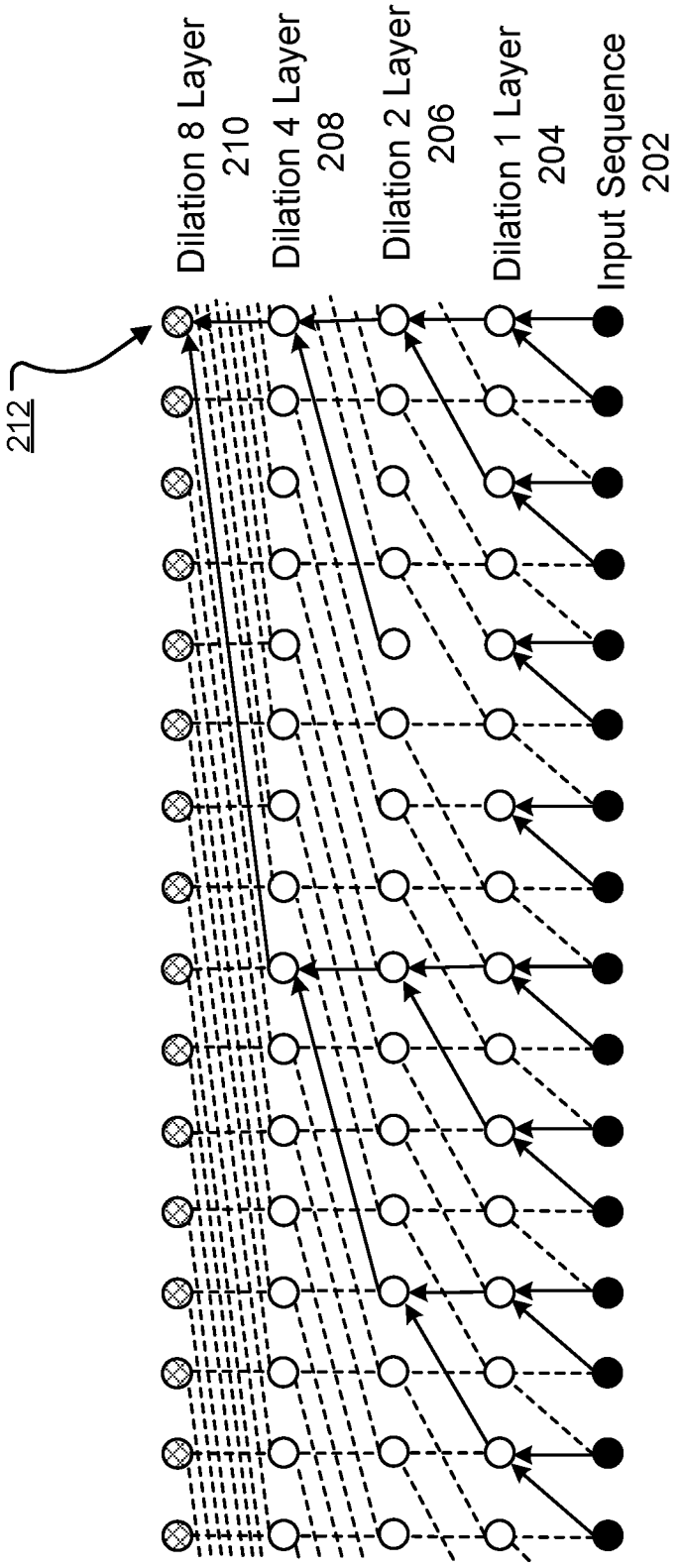
Dilation 8 Layer
210
Dilation 4 Layer
208
Dilation 2 Layer
206
Dilation 1 Layer
204
Input Sequence
202
212
200

FIG. 2

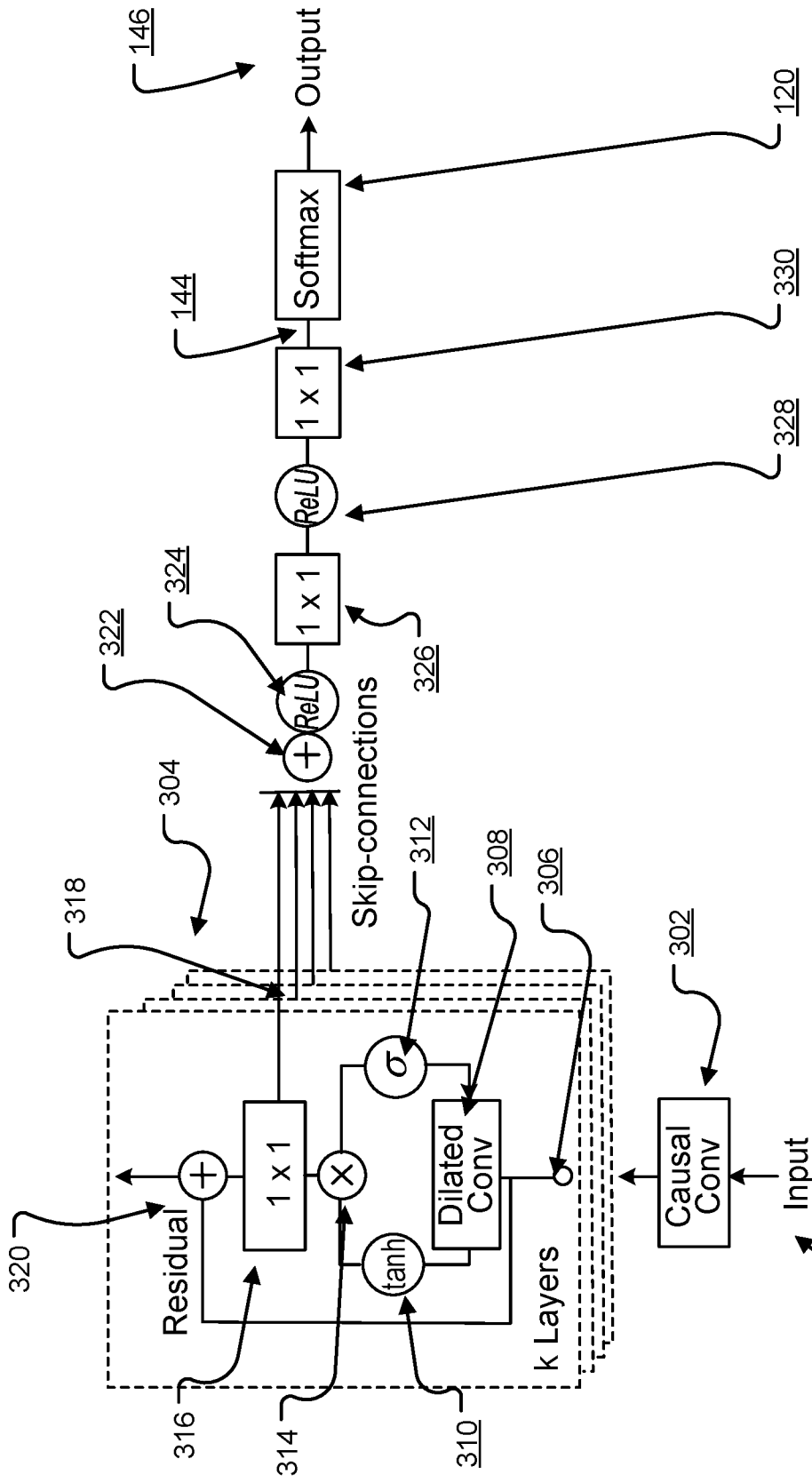
146
Output
120
Softmax
144
330
1 x 1
328
ReLU
324
322
1 x 1
326
ReLU
304
Skip-connections
312
308
306
318
302
σ
Dilated Conv
Causal Conv
Input
300
320
Residual
1 x 1
tanh
k Layers
316
314
310
142

FIG. 3

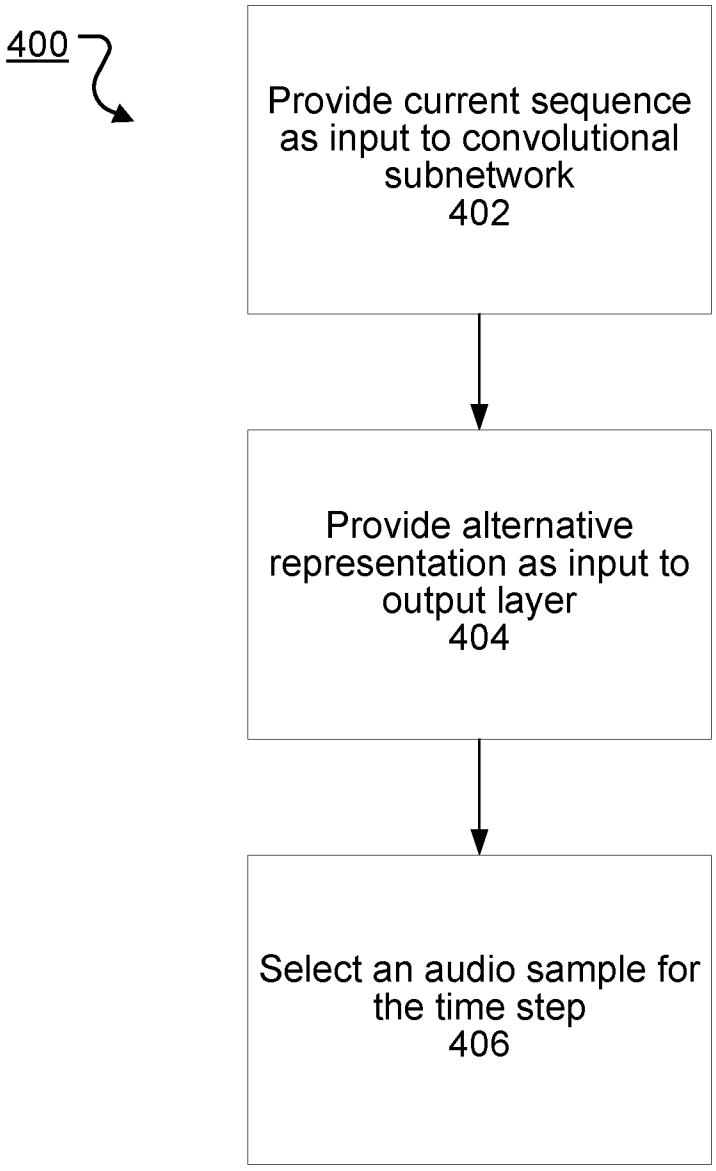
400
Provide current sequence as input to convolutional subnetwork
402
Provide alternative representation as input to output layer
404
Select an audio sample for the time step
406

FIG. 4

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2017/050320

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06N3/04
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

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	CA 2 810 457 Al (PENN GERALD BRADLEY [CA] ; JIANG HUI [CA] ; ABDELHAMID OSSAMA ABDELHAMID) 25 September 2014 (2014-09-25) abstract paragraph [0006] paragraph [0015] - paragraph [0022] -----	1-37
A	Aaron Van Den Oord ET AL: "Pi xel Recurrent Neural Networks" , , 19 August 2016 (2016-08-19) , pages 1-11 , XP055435831 , Retri eved from the Internet: URL: https ://arxi v.org/pdf/1601 .06759 .pdf [retri eved on 2017-12-18] abstract secti on "3.3. Resi dual Connecti ons" secti on "3.2. Di agonal Bi LSTM" ----- -/- ·	1-37

X Further documents are listed in the continuation of Box C.

X 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" documentwhich may throw doubts on priority claim(s) orwhich 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
18 December 2017	02/01/2018

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

3

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

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X,P	AARON VAN DEN OORD ET AL: "WaveNet: A Generative Model for Raw Audio", ARXIV.ORG, CORNELL UNIVERSITY LIBRARY, 201 OLIN LIBRARY CORNELL UNIVERSITY ITHACA, NY 14853, 12 September 2016 (2016-09-12), XP080725972, the whole document -----	1-37

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/US2017/050320

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
CA 2810457 A1	25-09-2014	NONE	