



(51) International Patent Classification:

G06N 3/06 (2006.01)

G06N 3/04 (2006.01)

G06F 17/16 (2006.01)

G10L 21/0208 (2013.01)

G06N 3/02 (2006.01)

(21) International Application Number:

PCT/US2022/026416

(22) International Filing Date:

26 April 2022 (26.04.2022)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

63/179,762

26 April 2021 (26.04.2021)

US

(71) Applicant: THE TRUSTEES OF DARTMOUTH COL-

LEGE [US/US]; 11 Rope Ferry Road, Room 6210, Hanover, New Hampshire 03755 (US).

(72) Inventor: ODAME, Kofi; 1 South Street, Apt. 303,

Hanover, New Hampshire 03755 (US).

(74) Agent: VOCK, Curtis et al.; Lathrop GPM LLP, 6640

Gunpark Drive, Suite 200, Boulder, Colorado 80301 (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, IT, JM, 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, 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, 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).

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))

(54) Title: LOW POWER ANALOG CIRCUITRY FOR ARTIFICIAL NEURAL NETWORKS

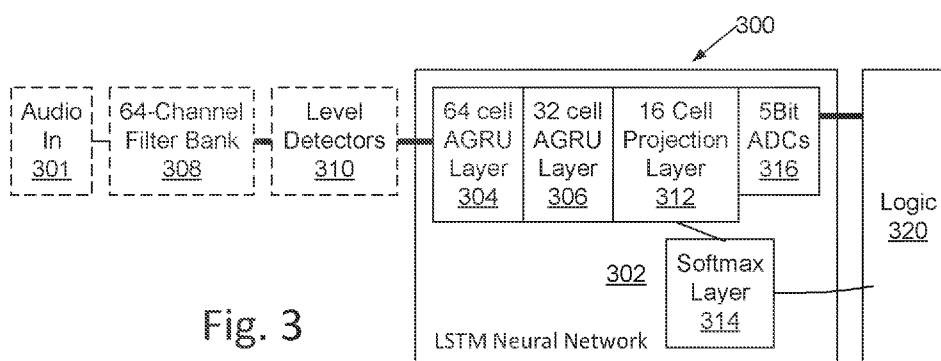


Fig. 3

(57) Abstract: A low power analog Long Short-Term Memory (LSTM) recurrent neural network has an input layer, an array of Adaptive Filter Unit for Analog LSTM, a linear projection layer, and an output layer. The output layer has multiple nonlinear amplifiers, a nonlinear element with a sigmoidal input-output characteristic function, and a time-constant adjustable, nonlinear, low pass filter that provides the memory function of the LSTM. The LSTM memory is used with mismatch-robust weights determined by learning by computation of optimal weights values, wherein the objective function minimizes misdetection probability, and used to process a signal to detect events.

LOW POWER ANALOG CIRCUITRY FOR ARTIFICIAL NEURAL NETWORKS

Claim to Priority

[001] The present document claims priority to U.S. Provisional Patent Application 63/179,762 filed 26 April, 2021. The entire contents of the aforementioned provisional application are incorporated herein by reference.

Background

[002] There are many applications for devices in body-area networks and internet-of-things (IoT) devices where both very low power consumption and accurate classification of received input data is desirable. Consider as an example, “biostamp” sensing devices that may be glued to skin of a subject and are configured to be always-on to sense physiological parameters of the subject, classify them, and transmit classification results to a body area network host device; biostamp devices have little room for batteries and must be able to operate for days between recharges so power budgets are quite small—in BioStamp® (trademark of Medidata, New York, NY, subsidiary of Dassault Systems, Vélizy-Villacoublay, France) or equivalent devices intended for wearing for a week or more between recharges, the power budget may be on the order of microwatts. Another example is security cameras and other always-on sensors that may be embedded into devices like cars and safes that may need to sound alarms or report through IoT devices when classified sensor data indicates a particular event.

[003] There are several known types of trainable classifiers used for artificial intelligence (AI) that can classify events such as those detected by always-on sensors, among these are electronic neural networks. Typically, electronic neural networks contain one or more layers of simulated “neurons”, where each “neuron” operates by performing a series of multiply-accumulate (MAC) operations, evaluating: $\sum_{i=1}^n p_i input_i$ for parameters, or weights, p_i that may be positive or negative, the summation is often followed by comparing the summed total to a threshold and/or limits to determine an output. Deep or multilayer neural networks may have several layers of “neurons,” where the outputs of neurons for each of the layers become inputs to the next layer; outputs of a final layer become outputs of the neural network.

[004] In biostamp and IoT applications, a “long short-term memory” (LSTM), deep, neural network may serve for real-time classification of sensor data.

[005] Many neural network integrated circuits operate by passing sensor data through an analog front end (AFE) and an analog-to-digital converter (ADC) before performing the MAC operations and thresholding operations in a digital processor. These circuits consume considerable power in the ADC and digital processor, in part because sensor data must be sampled at least at twice the maximum frequency of the sensor data.

[006] In some low-power, wearable systems, an energy-harvesting device may supplement energy stored in a battery or ultracapacitor with energy harvested from wearer movements, ambient light, or temperature differences between skin temperature and ambient temperature.

[007] Gated recurrent units (GRU) are a known type of multilayer neural network where some outputs of a layer are also inputs to that same layer, providing that layer with a memory function.

Summary

[008] A low power analog Long Short-Term Memory (LSTM) recurrent neural network has an input layer, an array of analog gated recurrent units (AGRUs), a linear projection layer, and an output layer. The output layer has multiple nonlinear amplifiers, a nonlinear element with a sigmoidal input-output characteristic function, and a time-constant adjustable nonlinear low-pass filter. The LSTM memory is used with mismatch-robust weights determined by learning by computation of optimal weights values, wherein the objective function minimizes misdetection probability, and used to process a signal to detect events.

Brief Description of the Figures

[009] Fig. 1 is a block diagram of a sensor system incorporating a battery with an optional energy-harvesting device, sensors, input circuitry, an LSTM artificial neural network, ADCs, digital output circuitry, micropower logic, and a low-power digital radio configured for reporting classified events to a host device.

[0010] Fig. 2 is a block diagram illustrating how a GRU is modified with a low-pass filter.

[0011] Fig. 3 is a block diagram of the analog LSTM artificial neural network, with an optional filterbank configured such that the LSTM artificial neural network can analyze sounds received by a microphone.

[0012] Fig. 4 illustrates simplification of a GRU neuron to an analog GRU (AGRU) neuron using a low-pass filter.

[0013] Fig. 5A illustrates a GRU input stage circuit including vector matrix multiplier.

[0014] Fig. 5B is a schematic diagram illustrating an envelope detector circuit as used in a GRU.

[0015] Fig. 5C is a schematic diagram illustrating circuitry for ternarized weight multiplication for an analog vector matrix multiplier (VMM).

[0016] Fig. 5D is a schematic diagram illustrating a sigmoidal functional block that serves as (F) functions in the AGRU.

[0017] Fig. 5E is a schematic diagram illustrating a nonlinear low pass filter used in the simplified AGRU.

[0018] Fig. 5F is a schematic diagram of an AGRU cell illustrating how the blocks of Fig. 5A, 5C, 5D, and 5E are used.

[0019] Fig. 6A is a block diagram of an alternative analog LSTM neural network with pyknogram filters and Adaptive Filter Unit for Analog LSTM (AFUA) neuron cells adapted for spoken keyword recognition.

[0020] Fig. 6B is a block diagram of a channel of a pyknogram filter used with a microphone and the analog LSTM neural network for spoken keyword recognition.

[0021] Fig. 7 is a simplified schematic diagram illustrating implementation of an AFUA) implementation of an analog recurrent unit using an analog low-pass filter in the memory path.

Detailed Description of the Embodiments

[0022] To avoid needing a power-consumptive AFE or ADC, we believe a deep neural network may operate directly on the sensor's output in the analog domain. Analog spiking neural networks like the Neurogrid, are an appealing option; even though they were not originally designed for the perceptron-type "artificial" neural networks (ANNs) that are used for deep learning, they can implement deep neural networks, either by converting a trained deep ANN into a spiking neural network

(SNN), or by building a deep SNN and then training it with a variant of backpropagation. Unfortunately, SNNs can only process data encoded as a train of spikes. Converting a conventional sensor output into this format requires a spike train generator, which, in an audio application, would consume most or all of the power budget in ultra-low power devices.

[0023] Non-spiking analog neural networks are typically built around a vector-matrix multiplier (VMM) core. This is a crossbar switch matrix that uses the fundamental analog circuit phenomena of Ohm's law and Kirchhoff's current law to implement the multiply-accumulate (MAC) operation with spectacular efficiency. VMM-based analog deep neural networks have been designed, built and tested on simple image classification tasks. Also, analog recurrent neural networks were tested on trajectory following problems. The results have been promising, although LSTM recurrent neural networks have not previously been demonstrated in analog hardware.

System

[0024] An example biostamp-type wearable device 100 is illustrated in Fig. 1. The sensor system incorporates an optional energy-harvesting device 102 coupled to charge a battery 104 that in turn powers the device, at least one sensor 112, 114 such as accelerometers, microphones, temperature sensors, gyroscopic sensors, moisture sensors, and other sensors for biologically significant parameters, input circuitry 116 adapted to process sensor outputs and prepare them for admission to an LSTM artificial neural network 118, ADC's 120, and digital output circuitry 122, micropower logic which may in some embodiments include a low-power processor and memory 124, and a low-power digital radio 126 configured for reporting classified events to a host device (not shown) for further processing. In a particular embodiment where at least one sensor 112 is a microphone, input circuitry 116 includes a bank of 64 low-power analog bandpass filters each feeding a level detector, with the level detector outputs feeding inputs of the LSTM artificial neural network 118.

Analog LSTM

[0025] Our available power budget is quite low, particularly when using an LSTM neural network in a biostamp application. To minimize physical obtrusiveness, it must be the approximate size and weight of a band-aid. To minimize disruption of the users' daily behaviors, it should not require battery changes or battery charging for

several weeks. These requirements taken together mean that the device must rely on energy harvesting, which can reliably provide $1\ \mu\text{W}$ or less of power.

[0026] In an embodiment 300 (Figure 3), our LSTM neural network 302 incorporates a ternarized-weight neural network 302 with a 64-dimensional input feature of filterbank energies provided by a filter bank 308 and envelope detector 310, followed by two unidirectional analog GRU (AGRU) layers 304, 306 of 64 and 32 units, respectively. Filter bank 308 receives an audio input 301, typically from a microphone, and, with the envelope detector 310, performs spectral analysis of the audio input; the spectrally analyzed output of the envelope detector 310 is fed to the first AGRU layer 304. Most embodiments use low-resolution weights; in this embodiment, three weights of -1, 0, and +1 are used in the AGRU layers of the neural network. In an alternative five-level-weight embodiment, weights of -1.0, -0.5, 0.0, +0.5, and +1.0 are used. In yet other embodiments, weights may have 7 levels, or another low-resolution set of possible weights.

[0027] After the two AGRU layers 304, 306, the next layer of the neural network is a 16-unit linear projection layer 312 for dimensionality reduction. The final output layer 314 is a softmax function. To provide input to a downstream automatic speech recognition (ASR) engine, when an event is detected the projection layer 312, the event is optionally digitized with an array of logarithmic 5-bit ADCs 316 for further speech processing. The digitized outputs of the ADCs and output of the output layer 314 are fed to a logic unit 320 that may incorporate a microprocessor, and which may have been held idle until the event was detected. This data representation eases energy requirements of the ADCs that are powered only when an event is detected by the LSTM network, while maintaining high ASR accuracy in processing subsequent voice commands or sounds of medical interest such as coughs and wheezes. The preamplifier consumes 500 nW. The analog LSTM consumes $2\ \mu\text{W}$ (assuming $0.1\ \text{pJ/computation}$). Digitization is achieved with slow (50 Sa/s), low resolution, successive approximation (SAR) ADCs, which consume less than 1 nW (assuming a conservative $20\ \text{fJ/step}$).

[0028] To implement a state-of-the-art LSTM for cough or wheeze detection using a microphone as a sensor, or a similar event detection problem, the hardware must execute an equivalent about 85 million vector-matrix multiplications and over 70 thousand Hadamard product multiplications per second. Conventional analog VMMs require about $0.1\ \text{pJ}$ per computation, while the element-wise multiplications of a Hadamard product consumes two to seven pJ per computation.

Analog mismatch

[0029] Two nominally-identical analog functional blocks will inherently be mismatched, due to nonidealities in the semiconductor fabrication process. Reducing mismatch ultimately results in increased power consumption, which defeats the purpose of implementing an analog LSTM in the first place. Theoretically, a neural network could incorporate analog nonidealities into its learned model, but there are some practical challenges. Backpropagation, the most widely-used learning algorithm for neural networks, relies on accurate information about the gradients of the network's activation functions. With mismatch, each of these gradients would have to be measured empirically, rather than calculated analytically.

[0030] To overcome this, characterization tests and parameter extraction is performed for every manufactured chip, on each instance of each activation function in the neural network. The derivative of each individual measured function is used in a standard backpropagation procedure for stochastic gradient descent. The customized learned weights are then programmed onto the chip under test.

[0031] We can avoid having to fully characterize each individual chip by training them via error descent. Generally, the idea is to apply a random test change to the network weights of a fabricated chip, and then observe how its output responds. This gives a crude measure of the error gradient, against which the weights can be updated. With many iterations of this process, the weights will converge, but only slowly (between $O(P)$ and $O(P^{1/2})$ per epoch, where P is the number of network parameters). The scheme is therefore unsuitable for training large numbers of systems on the large datasets that are used in LSTM networks.

Innovative Nonlinear dynamical approach

[0032] We will apply the nonlinear dynamical systems approach to implement the LSTM network. When others have considered the design of analog neural networks, it usually involves implementing (or approximating) each discrete mathematical operation with a suitable analog circuit component (e.g. transconductance elements, Gilbert cell multipliers, PWL nonlinear devices from diodes and op-amps, etc.). These are reasonable ways of synthesizing nonlinear processors, but they yield comparatively inefficient implementations.

[0033] To achieve high functional density and efficiency, the nonlinear dynamical system must be considered as a whole. We can characterize the processing function in nonlinear dynamical system terms. For example, does it have a limit cycle? Does it display hysteresis, or multiple steady-state behavior? Once the fundamental dynamics have been identified, we can then create a circuit that also displays these properties and that is topologically equivalent to the original dynamical system. A type of LSTM called the gated recurrent unit (GRU) is illustrated in Fig. 2 and 5F. It exhibits a saddle-node bifurcation into a stable and unstable operation areas, and it also has an input- dependent characteristic time scale (analogous to the time constant in a linear time invariant system) via the ‘z’ signal. As we show in the preliminary results section, careful consideration of these and other properties results in an efficient circuit realization that does not require explicit implementation of every discrete mathematical operation in the GRU state update equation.

[0034] In addition to the system-level approach to efficiency, we also propose component-level changes to increase efficiency. In particular, we propose a novel VMM that is roughly 30 times more power-efficient than the state-of-the-art. The essential idea is to reduce the impedance of the VMM’s internal nodes by terminating the VMM’s accumulation line with a gm-boosted diode-connected transistor. Gm-boosting increases the transconductance (reduces the impedance) of the transistor by the factor of the loop gain. This reduction in impedance pushes parasitic poles to higher frequencies, allowing the VMM to maintain its operation speeds with lower power consumption.

Training with perturbed networks

[0035] A way to reduce the detrimental effects of noise or mismatch in a neural network is to average the predictions of a large number of these nonideal networks. This idea was demonstrated by training several weak linear analog classifiers, then presenting them each with test data, and finally combining their predictions via AdaBoost.

[0036] Our innovation applies the same general concept, but in a way that is computationally feasible for an LSTM network. The idea is to train a large collection of networks that all share weights, and that are all perturbed versions of the same nominal model. The amount and distribution of network perturbations mirror the process variation and device mismatch that would be observed in actual fabricated analog

circuits. Each training example is applied to a different perturbed network, and the updated weights are shared with the next, randomly-selected perturbed network. This allows the weights to converge in a way that is robust to analog circuit mismatch.

[0037] The procedure is essentially a variation of training with noise regularization, which has been shown to lead to convergence, and even improved performance, under certain conditions. Where noise regularization typically injects noise onto weights or inputs, our approach also adds noise onto the parameters that define activation functions.

[0038] We are systematically injecting noise to model the amount of mismatch that is expected from chip fabrication. We are trying to avoid overfitting not just to the data, but also to any given instance of the fabricated chip, within certain yield targets.

Preliminary results

[0039] We used our nonlinear design methodology to make some initial designs of a simpler GRU variant as illustrated in Fig. 4, where a GRU neuron is simplified to an analog GRU (AGRU) neuron using a low-pass filter to provide the recurrent memory function in each neuron or recurrent unit. It is able to exploit the inherent nonlinearities of analog circuits to achieve an extremely efficient implementation. There are fewer cascaded operations necessary to implement the simpler GRU, so the operations can each run a little slower and still meet latency requirements. This simplification, combined with a more efficient VMM, reduces the power consumption of each GRU layer by a factor of 50 to 100.

[0040] The reduction in power consumption does not alter the basic dynamics of the GRU, the simpler GRU exhibits a saddle-node bifurcation and a vector field that are similar to those of the original system.

[0041] In this simpler, but topologically equivalent system, there are an analog low-pass filter (labeled 'LPF') and a pair of nonlinear activation functions. The original state variables h , $\tilde{h}$, z are identified with the new state variables h^+ , $\tilde{h}^+$, z^+ , respectively. The block labelled 'LPF' (Fig. 4) is an analog current- mode low-pass filter with a z^+ -controllable time constant and, in an embodiment, has a schematic illustrated in Fig. 5E. Also, the activation function, f_{402} , is implemented with a diode-

connected transistor and a differential pair. Note: to avoid clutter, the weight matrix multiplications have been omitted in Figs. 4 and 5A-B and 5D-E.

Mismatch-robust training

[0042] We applied an early version of our mismatch-robust training scheme to a neural network with sigmoidal and tanh activation functions. Perturbed networks were created by randomly varying the slope and vertical and horizontal offsets of the activation functions. These random variations were sampled from normal distributions with standard deviations of 0.11 and means of 1 (slope) and 0 (offsets). The network was tested on a spoken digits dataset, with 2114 samples for training and 907 samples for validation.

Approach with Analog GRU network

[0043] Our hardware LSTM recurrent neural network is a ternarized-weight (or suitably low-precision weight) neural network with a 64-dimensional input feature of filterbank energies, followed by two unidirectional GRU layers of 64 and 32 units, respectively. The next layer is a 16-unit linear projection layer for dimensionality reduction. The output layer is a softmax function. We plan to do a hyperparameter exploration to improve on this initial model.

[0044] Simplified schematics of the major circuit components are shown in figs. 5A-5E.

[0045] Fig. 5A illustrates a GRU input stage. Bandpass filters and envelope detectors produce filterbank energies, which are multiply-accumulated by ternarized multiply-accumulate blocks 510 (Fig. 5F), illustrated as diamonds in Fig. 5A, and detailed in Fig. 5C, onto column lines as shown in Fig. 5F.

[0046] Fig. 5B illustrates an envelope detector, used to determine filterbank energies from filterbank outputs. The input current, I_{in} , is rectified by the diode-connected pMOS and then smoothed with a current-mode, analog, low-pass filter. Fig. 5C illustrates ternarized weight multiplication for an analog VMM. By switching the D_0 , D_1 switches appropriately, I_{in} is multiplied by the weight parameter, one of $\{-1, 0, 1\}$. Also shown in Fig. 5D, is a sigmoidal functional block used for functional blocks 402, 512 (Fig. 5F). The left-most pMOS is a common gate amplifier that provides gm-boosting to the cascoded nMOS pair. The impedance encountered by the $\omega_k I_k$ current

is reduced by a factor of the common gate amplifier gain. This pushes the parasitic pole of the VMM accumulation line to high frequencies, allowing the VMM to run with low power consumption. Sigmoidal behavior is derived from a combination of the single-endedness enforced by the nMOS input sink and the output differential pair stage. Fig. 5E details a nonlinear low pass filter 514. The input to the filter is the current corresponding to h_i . Also, the normally-fixed “time-constant” current is set by the z_i signal. This effectively acts as a knob that either allows or impedes updating of the state variable, h .

Monte Carlo Backpropagation

[0047] Statistical models on device mismatch, such as those provided by a semiconductor foundry, will be used to generate a different perturbed network for each update step of a stochastic gradient descent training scheme. The forward pass will be performed on the perturbed network, using low-precision weights. The backpropagation pass will use full-precision weights, or else a quantized backpropagation method. Also, gradients will be averaged over the set of perturbed networks that are used in a given mini-batch.

[0048] The steps of the perturbed network training scheme involve: (1) Monte Carlo analysis – based on statistical parameters that are empirically measured by the semiconductor fabrication foundry – will be performed on the transistor-level description of the neural network’s activation functions and other mathematical operators. Monte Carlo simulation samples the device matching and process variations and is a standard tool used in the semiconductor industry to accurately predict the variance and yield of a chip design. This analysis will yield information on the probability distribution of activation functions’ parameters (e.g. the slope of rectified linear units (ReLU) as known in the art of neural networks); (2) generating perturbed network models. A new, perturbed network will be generated upon the presentation of each training case. The activation functions of the network will each be perturbed from their nominal parameters, using the probability distributions derived from the previous step. To keep the algorithm computationally tractable, weights will be shared between perturbed networks; (3) Weight programming. The final step of the scheme is to program the learned weights onto a physical integrated circuit realization of the network. Since this is an analog chip implementation, it is in essence just another perturbed network, sampled from the same distribution that was used to learn the

programmed weights. Thus, the programmed chip should be automatically compensated against its inherent mismatch and variation.

Speech Recognition with Pyknoogram Filter and AFUA

[0049] We anticipate that the low-power neural network herein disclosed may prove useful in classifying a wide variety of sensor data, including motion sensor data for discrimination of walking or running from other movements, and in particular may serve as a classifier stage to classify audio that has been preprocessed into frequency domain or otherwise spectrally analyzed by a filter bank or fast-Fourier transform. We expect the low-power neural network to be able to classify preprocessed audio signals of many types, such as dog barks, cat meows, bird songs, human coughs or wheezes, heart murmurs, and speech.

[0050] In a particular embodiment 600, illustrated in Fig. 6A we have combined a low-power neural network with a 4-channel pyknoogram filterbank 604 to classify speech recorded by a microphone 602. In this embodiment, the pyknoogram filterbank 604 and amplitude detectors 606 preprocesses the speech by performing spectral analysis providing frequencies 608 and detected amplitudes to the low power LSTM neural network 610. The low-power LSTM neural network 610 detects and identifies ten spoken keywords with better than 90% accuracy with estimated power consumption of about $\frac{3}{4}$ microwatt when implemented in a 180 nanometer CMOS process.

Pyknoogram Filterbank

[0051] The pyknoogram filterbank 604 produces a dense, low dimensional version of a spectrogram. The pyknoogram filterbank 604 has multiple filter channels, in a particular embodiment only 4 channels provided sufficient spectral information for speech recognition. In other embodiments, additional channels may be provided. Each pyknoogram filterbank 604 channel has structure illustrated in Fig. 6B. First, a bandlimited signal, V_u , from microphone 602 is input to an adaptive bandpass filter 605 (Fig. 6B). Then, the adaptive filter attempts to minimize an error between its input V_u and its output V_w by adjusting its center frequency, f_c ; the error is minimized when f_c is tracking the input signal's most energetic frequency region. Thus, the center frequency, f_c , is an output "feature" of the pyknoogram analysis, as is an amplitude A_f . For voice inputs f_c corresponds to the speech formant.

[0052] In each channel of the pyknoqram filterbank, NMOS M_x is in the subthreshold regime with a drain current, I_x , given by

$$C_x \, dI_x/dt = (\kappa Gm/UT) \cdot (V_u - V_w) \cdot \text{sgn}(V_y) \cdot I_x,$$

where κ is the body-effect coefficient and U_T is a thermal voltage. I_x is also the bias current that controls the gain of the transconductance amplifiers, and hence the filter's center frequency. So, the pyknoqram filter channel continually adjusts I_x —and hence the center frequency—to minimize the $(V_u - V_w)$ error and thereby track the most energetic frequency region in the input speech signal.

Each channel of pyknoqram filter is configured to process a particular sub-band of input audio, in an embodiment by feeding audio to each pyknoqram filter through an appropriate bandpass filter, and in an alternative embodiment by limiting the range of control current I_x .

AGRU->AFUA for Speech

[0053] We can simplify a GRU with a simpler set of equations via the following manipulations. The sigmoid function of gives z_j a range of (0, 1), and extrema of this range reveals the basic mechanism of an update equation. For $z_j = 0$, the update equation is $h_j^{(t)} = \tilde{h}_j^{(t)}$. For $z_j = 1$, the update equation becomes $h_j^{(t)} = h_j^{(t-1)}$. Without loss of generality, we replace $(1 - z_j)$ with z_j (this merely inverts the logic of the update gate). So, replacing $(1 - z_j)$ and rearranging the update equation and recognizing similarity to a transfer function of a first-order low pass filter with a continuous-time form of:

$$\frac{\tau}{z_j(t)} \frac{dh_j}{dt} + h_j(t) = \bar{h}_j(t)$$

where $\tau = \Delta T$, the time step of the discrete-time system. The gating mechanics of the continuous- versus discrete-time update equations are equivalent, modulo the inverted logic: For $z_j(t) = 0$, this amounts to a low-pass filter with an infinitely large time constant, and $h_j(t)$ does not change. For $z_j(t) = 1$, this is a low-pass filter with a time constant of $\tau = \Delta T$. The ΔT time step is small relative to the GRU's dynamics.

[0054] Finally, if we translate the origins of both $h_j(t)$ and $\tilde{h}_j(t)$ to 1, then we can replace the tanh with a saturating function that has a range of (0, 2). Replacing tanh with $\min(\text{softplus}(\cdot), 2)$, translating the origin, and discarding the reset gate, we arrive at the Adaptive Filter Unit for Analog LSTM (AFUA) as illustrated in Fig. 7.

[0055] Frequency and amplitude outputs of the pyknoqram filterbank are fed to a trained low-power neural network as herein disclosed having 2264 ternary weight parameters, the weight parameters having been trained using a database of multiple speakers speaking digits, with 2114 samples for training and 907 samples for validation.

[0056] In this embodiment, we use as a recurrent unit an Adaptive Filter Unit for Analog LSTM (AFUA) that implements the equations:

$$z_j(t) = \min(\text{soft}([Wz_x]_j + [U_z(h(t) - 1)]_j), 1)$$

$$\tilde{h}_j(t) = \min(\text{soft}([W_x]_j + [U(h(t) - 1)]_j), 2)$$

$$\frac{\tau}{z_j(t)} \frac{dh_j}{dt} = \bar{h}_j(t) - h_j(t),$$

where we have also replaced the sigmoid with a clipped softplus. The benefit of the softplus over the sigmoidal function is that we can implement it as a diode-connected transistor, which costs minimal area and zero power.

[0057] The AFUA is a type of continuous-time GRU. But while the GRU contains 3 Hadamard multiplications, the AFUA contains none. Further, the AFUA has no reset gate.

[0058] Finally, as a circuit embodiment of an embodiment of an AFUA illustrated in Fig. 7 shows, the AFUA avoids the overhead of operational amplifiers, current/voltage converters and internal digital/analog converters found in other analog LSTM implementations. Fewer operations and smaller overhead mean less power consumption.

[0059] In the embodiment of Fig. 7, drain currents of M_z and $M_{\tilde{h}}$ are $I_z = \min(\text{soft}([WzI_x] + [U_z(I_h - I_{unit})]), I_{unit})$, $I_{\tilde{h}} = \min(\text{soft}([WI_x] + [U(I_h - I_{unit})]), 2I_{unit})$.

Also, from the translinear loop principle, current I_h is defined by:

$$\frac{C_z U_T}{k I_{unit}} \frac{I_{unit}}{I_z} \frac{dI_h}{dt} = I_{\bar{h}} - I_h$$

These equations represent the current-mode representations of the AFUA. When I_z is close to zero, the hidden state (represented by I_h) changes slowly, and the AFUA can retain long-term memory. When I_z is large, the state's rate of change increases, and the AFUA replaces its memory with newly arrived information via $I_{\tilde{h}}$.

[0060] In the embodiment of Fig. 7, each AFUA 700 recurrent unit cell has multiple inputs some of which 702 come from either a prior layer or the pyknogram filter, and some of which 704 come from low pass filters of the same or other AFUA recurrent cells of the same layer. Each input is buffered and brought through ternary-weight multiply-accumulate circuitry 706 onto accumulator column lines 708, 710. Each accumulator column line 708, 710 is coupled through a softmax function block 712, 714, implemented with diode-coupled transistors, to an analog low-pass filter 716. Analog low pass filter output is tapped 718 to drive the next layer of the analog LSTM neural network, and to provide delayed feedback into the current layer thereby providing the long-short term memory function.

Results

[0061] We evaluated the AFUA against the classical LSTM in a 10-word (spoken digits) keyword spotting task based on the Google Speech Commands dataset. We implemented the following network architecture in Tensorflow: two 16-unit recurrent layers 612, 614, of sixteen AFUA cells (each recurrent layer in alternative embodiments includes multiple AFUA, AGRU, or LSTM cells receiving the layer's inputs—such as inputs to the neural network or outputs of the prior recurrent layer—and outputs of the same layer providing recurrency) followed by a 10-unit linear projection layer 616 with ReLU units, and a 10-unit softmax output layer 618. The train/validation/test data split was 53/17/30. We used the ADAM optimizer, mini-batch sizes of 1024 and trained for 2000 epochs while using ternary weights. This is only an example we experimented with, we expect a wide variety of neural networks with two or more layers of AFUA or AGRU cells should suffice for decoding a repertoire of ten spoken keywords; larger vocabularies may require additional AFUA or AGRU neuron cells, additional projection layer cells, and additional softmax layer cells to accommodate.

[0062] The AFUA performs similarly to AGRU and LSTM. The AFUA's advantage is fewer inputs and fewer operations per neuron, the AFUA's computational complexity is up to 100× less than that of other approaches. Also, the AFUA is a smaller model, as illustrated by its parameter count. Finally, the AFUA implemented in a 0.18 μm 1.8 V process with $I_{\text{unit}} = 0.2 \text{ nA}$ would consume 0.76 μW . This power consumption meets the constraints of a self-powered sensor

Combinations

[0063] The features and circuits herein described can be implemented in multiple combinations and variations. Among combinations anticipated by the inventors are:

[0064] A low power analog Long Short-Term Memory (LSTM) recurrent neural network signal processing device designated A including input circuitry, and an array, the array including at least one layer of a plurality of recurrent units selected from the group consisting of Adaptive Filter Unit for Analog LSTM (AFUA) and Analog Gated Recurrent Units (AGRU), the array configured to receive input signals from the input circuitry; and a linear projection layer; and an output layer. The recurrent units each include a vector matrix multiplier array (VMM) circuitry configured to sum a plurality of products weights times inputs of the recurrent units.

[0065] A low power analog LSTM recurrent neural network signal processing device designated AA including the device designated A, wherein: the input circuitry is configured to couple at least one preprocessed input signal from sensors to the array.

[0066] A low power analog LSTM recurrent neural network signal processing device designated AB including the device designated A or AA, wherein the input circuitry includes: a plurality of analog bandpass filters coupled to a plurality of envelope detectors, such that a first end of a plurality of the analog bandpass filters is coupled to an input and a second end of a plurality of the analog bandpass filters is coupled to an input of an envelope detector, and outputs of the plurality of envelope detectors couple to the array.

[0067] A low power analog LSTM recurrent neural network signal processing device designated AC including the device designated AB, wherein the envelope detector comprises a rectifier and a current-mode low pass filter, wherein the rectifier is coupled to the current-mode low pass filter.

[0068] A low power analog LSTM recurrent neural network signal processing device designated AD including the device designated A, AA, AB, or AC, wherein: the linear projection layer reduces a signal dimensionality of a prior layer of recurrent units by applying an adjustable weight to each output of the prior layer of recurrent units to form weight-adjusted signals, and then summing the weight-adjusted signals.

[0069] A low power analog LSTM recurrent neural network signal processing device designated AE including the device designated A, AA, AB, AC, or

AD, wherein: the recurrent units of the array each receive a plurality of analog signals and multiply each analog signal by an adjustable low bit-width resolution weight value.

[0070] A low power analog LSTM recurrent neural network signal processing device designated AF including the device designated AE, wherein: each low bit-width resolution weight is ternarized to one of $\{-1, 0, 1\}$.

[0071] A low power analog LSTM recurrent neural network signal processing device designated AF including the device designated A, AA, AB, AC, AE, AF, or AE, wherein: the recurrent units of the array are AGRUs, and the AGRUs further comprise sigmoidal functional blocks and a low pass filter having an adjustable time constant.

[0072] A low power analog LSTM recurrent neural network signal processing device designated AG including the device designated A, AA, AB, AC, AD, or AE, wherein the input circuitry comprises a plurality of pyknoqram filters.

[0073] A low power analog LSTM recurrent neural network signal processing device designated AH including the device designated AG, wherein each pyknoqram filter comprises an adaptive bandpass filter having a center frequency, the adaptive bandpass filter configured to track a frequency of a strongest signal within a frequency range.

[0074] A low power LSTM recurrent neural network signal processing device designated AJ including the device designated A, AA, AB, AC, AD, AE, AG, or AH \, wherein the recurrent units of the array are AFUAs.

[0075] A low power LSTM recurrent neural network signal processing device designated AK including the device designated AJ, where each AFUA recurrent unit cell comprises a ternary-weight multiply-accumulate circuitry coupled through a softmax function to an analog low-pass filter.

[0076] A method for detecting events designated B, including: using a method comprising learning mismatch-robust weights learning for a low power analog LSTM recurrent neural network signal processing device incorporating computation of optimal weights values, wherein an objective function is to minimize a misdetection probability; where the LSTM recurrent neural network includes: an array comprising a plurality of Adaptive Filter Unit for Analog LSTM (AFUAs) analog recurrent units, a linear projection layer, and an output layer; and operating the LSTM recurrent neural network with the computed optimal weights values to process a signal to detect the events; and wherein the AFUAs comprise a low pass filter array and a vector matrix

multiplier array (VMM) comprising circuitry configured to sum a plurality of products of multiplying weights times inputs of the AFUAs.

[0077] A method designated BA including the method designated B, wherein the computed optimal weights values are computed using a multi-iteration Monte Carlo Backpropagation method adapted to minimize effects of component mismatch in the AFUAs.

[0078] A method designated BB including the method designated BA, wherein the multi-iteration Monte Carlo Backpropagation method includes: using statistical parameters applied to a transistor-level description of the LSTM neural network's activation functions and mathematical operators to simulate matching and process variations to give a probability distribution of activation function parameters and generating perturbed network models; and perturbing activation functions of the LSTM neural network from the probability distributions of activation function parameters.

[0079] A method designated BC including the method designated B, BA, or BB, further including: generating a set of perturbed networks from statistical models on fundamental device-level mismatch, and iteratively computing activation function parameters for each of the set of perturbed networks, wherein low-precision weights are used during forward pass calculations and full-precision weights are used during backpropagation pass calculations.

[0080] A method of detecting spoken keywords designated C including: filtering audio to determine a spectral analysis of an audio signal; and providing the spectral analysis of the audio signal to the method for detecting events designated B, BA, BB, or BC.

[0081] Changes may be made in the above methods and systems without departing from the scope hereof. It should thus be noted that the matter contained in the above description or shown in the accompanying drawings should be interpreted as illustrative and not in a limiting sense. The following claims are intended to cover all generic and specific features described herein, as well as all statements of the scope of the present method and system, which, as a matter of language, might be said to fall therebetween.

CLAIMS

1. A low power analog Long Short-Term Memory (LSTM) recurrent neural network signal processing device comprising:
input circuitry, and
an array, the array comprising at least one layer of a plurality of recurrent units selected from the group consisting of Adaptive Filter Unit for Analog LSTM (AFUA) and Analog Gated Recurrent Units (AGRU), the array configured to receive input signals from the input circuitry, and
a linear projection layer, and
an output layer;
wherein the recurrent units each comprise a vector matrix multiplier array (VMM) comprising circuitry configured to sum a plurality of products weights times inputs of the recurrent units.
2. The low power analog LSTM recurrent neural network signal processing device of claim 1, wherein:
the input circuitry is configured to couple at least one preprocessed input signal from sensors to the array.
3. The low power analog LSTM recurrent neural network signal processing device of claim 2, wherein the input circuitry comprises:
a plurality of analog bandpass filters coupled to a plurality of envelope detectors, such that a first end of a plurality of the analog bandpass filters is coupled to an input and a second end of a plurality of the analog bandpass filters is coupled to an input of an envelope detector, and outputs of the plurality of envelope detectors couple to the array.
4. The low power analog LSTM recurrent neural network signal processing device of claim 3, wherein the envelope detector comprises a rectifier and a current-mode low pass filter, wherein the rectifier is coupled to the current-mode low pass filter.
5. The low power analog LSTM recurrent neural network signal processing device of claim 1, wherein:

- the linear projection layer reduces a signal dimensionality of a prior layer of recurrent units by applying an adjustable weight to each output of the prior layer of recurrent units to form weight-adjusted signals, and then summing the weight-adjusted signals.
6. The low power analog LSTM recurrent neural network signal processing device of claim 1, wherein:
the recurrent units of the array each receive a plurality of analog signals and multiply each analog signal by an adjustable low bit-width resolution weight value.
 7. The low power analog LSTM recurrent neural network signal processing device of claim 6, wherein:
each low bit-width resolution weight is ternarized to one of $\{-1, 0, 1\}$.
 8. The low power analog LSTM recurrent neural network signal processing device of claim 1, wherein:
the recurrent units of the array are AGRUs, and the AGRUs further comprise sigmoidal functional blocks and a low pass filter having an adjustable time constant.
 9. The low power analog LSTM recurrent neural network of claim 2, wherein the input circuitry comprises a plurality of pyknoogram filters.
 10. The low power analog LSTM recurrent neural network of claim 9, wherein each pyknoogram filter comprises an adaptive bandpass filter having a center frequency, the adaptive bandpass filter configured to track a frequency of a strongest signal within a frequency range.
 11. The low power LSTM recurrent neural network of claim 10, wherein the recurrent units of the array are AFUAs.
 12. The low power LSTM recurrent neural network of claim 11, where each AFUA recurrent unit cell comprises a ternary-weight multiply-accumulate circuitry coupled through a softmax function to an analog low-pass filter.

13. A method for detecting events, comprising:
 - using a method comprising learning mismatch-robust weights learning for a low power analog LSTM recurrent neural network signal processing device comprising:
 - computation of optimal weights values, wherein an objective function is to minimize a misdetection probability;
 - where the LSTM recurrent neural network comprises: an array comprising a plurality of Adaptive Filter Unit for Analog LSTM (AFUAs) analog recurrent units, a linear projection layer, and an output layer;
 - and operating the LSTM recurrent neural network with the computed optimal weights values to process a signal to detect the events; and
 - wherein the AFUAs comprise a low pass filter array and a vector matrix multiplier array (VMM) comprising circuitry configured to sum a plurality of products of multiplying weights times inputs of the AFUAs.
14. A method of claim 13, wherein the computed optimal weights values are computed using a multi-iteration Monte Carlo Backpropagation method adapted to minimize effects of component mismatch in the AFUAs.
15. A method of claim 14, wherein the multi-iteration Monte Carlo Backpropagation method comprises:
 - using statistical parameters applied to a transistor-level description of the LSTM neural network's activation functions and mathematical operators to simulate matching and process variations to give a probability distribution of activation function parameters and generating perturbed network models; and
 - perturbing activation functions of the LSTM neural network from the probability distributions of activation function parameters.
16. A method of claim 13, further comprising:
 - generating a set of perturbed networks from statistical models on fundamental device-level mismatch, and
 - iteratively computing activation function parameters for each of the set of perturbed networks, wherein low-precision weights are used during forward pass calculations and full-precision weights are used during backpropagation pass calculations.

17. A method of detecting spoken keywords comprising:
filtering audio to determine a spectral analysis of an audio signal; and
providing the spectral analysis of the audio signal to the method for detecting events of
claim 13.

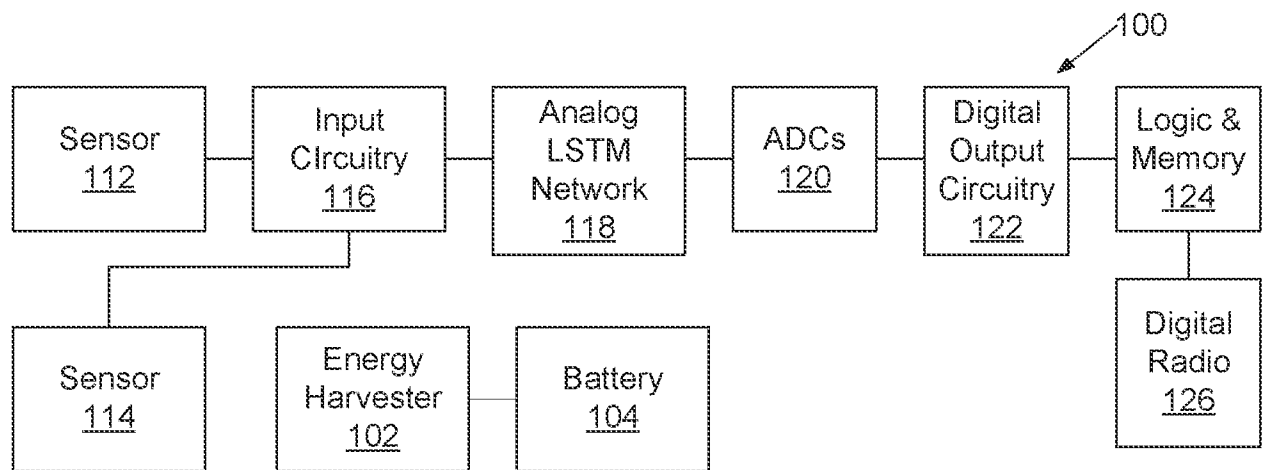


Fig. 1

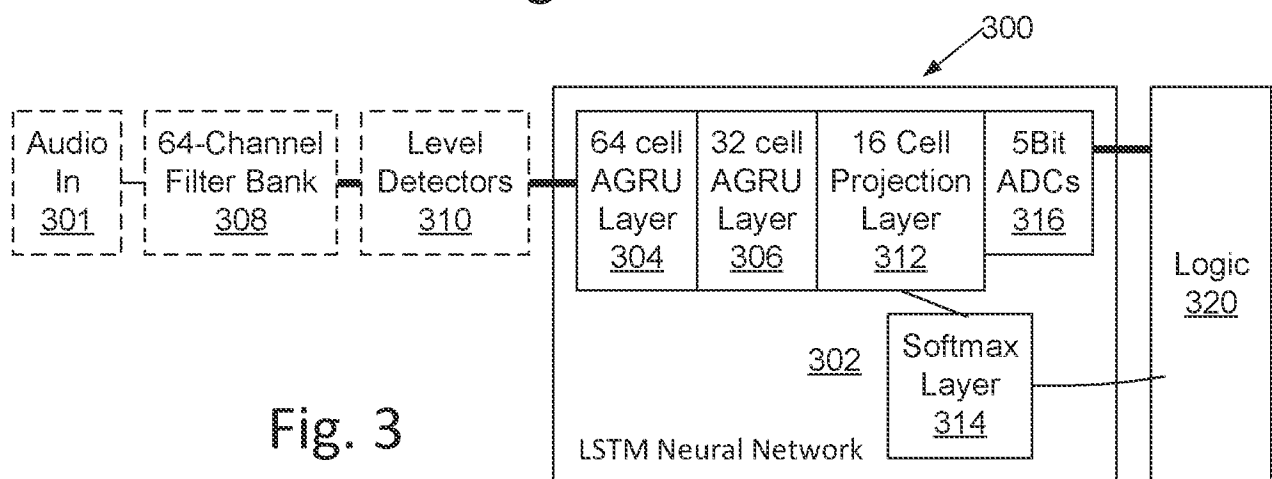


Fig. 3

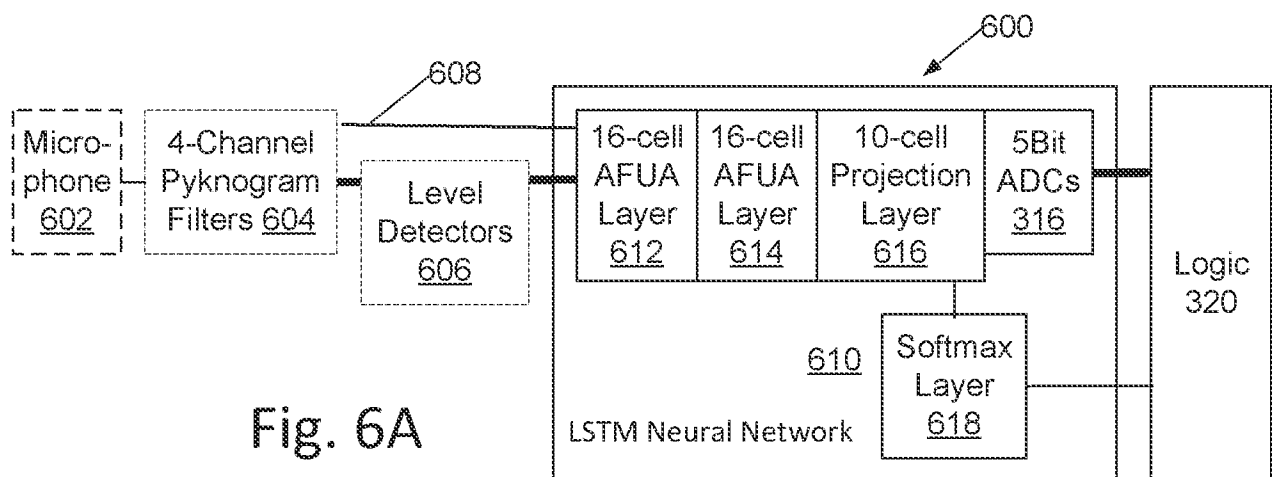


Fig. 6A

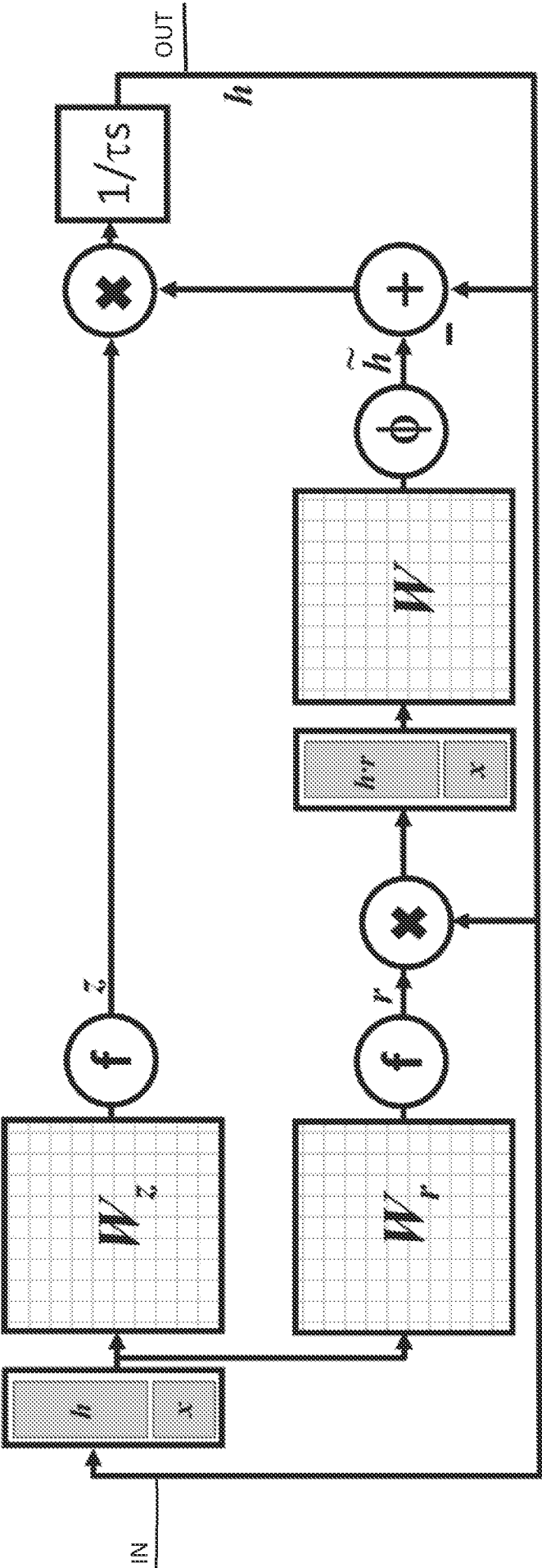
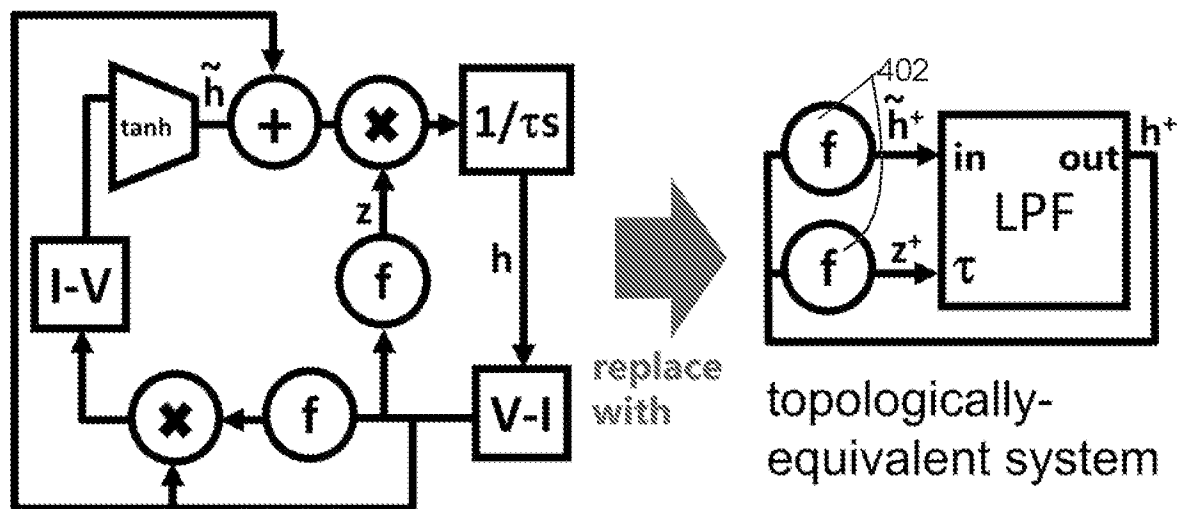


Fig. 2



original GRU
dynamical system

Fig. 4

from vector matrix multiplier

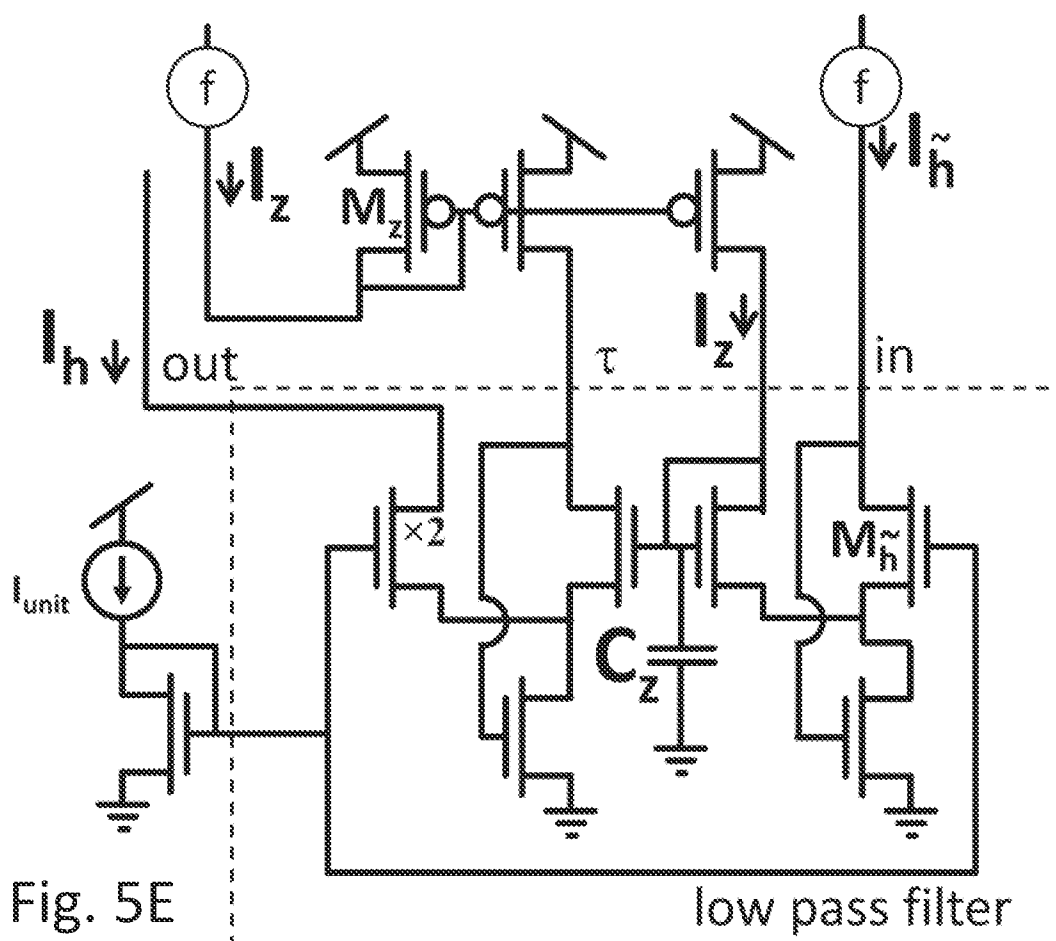


Fig. 5E

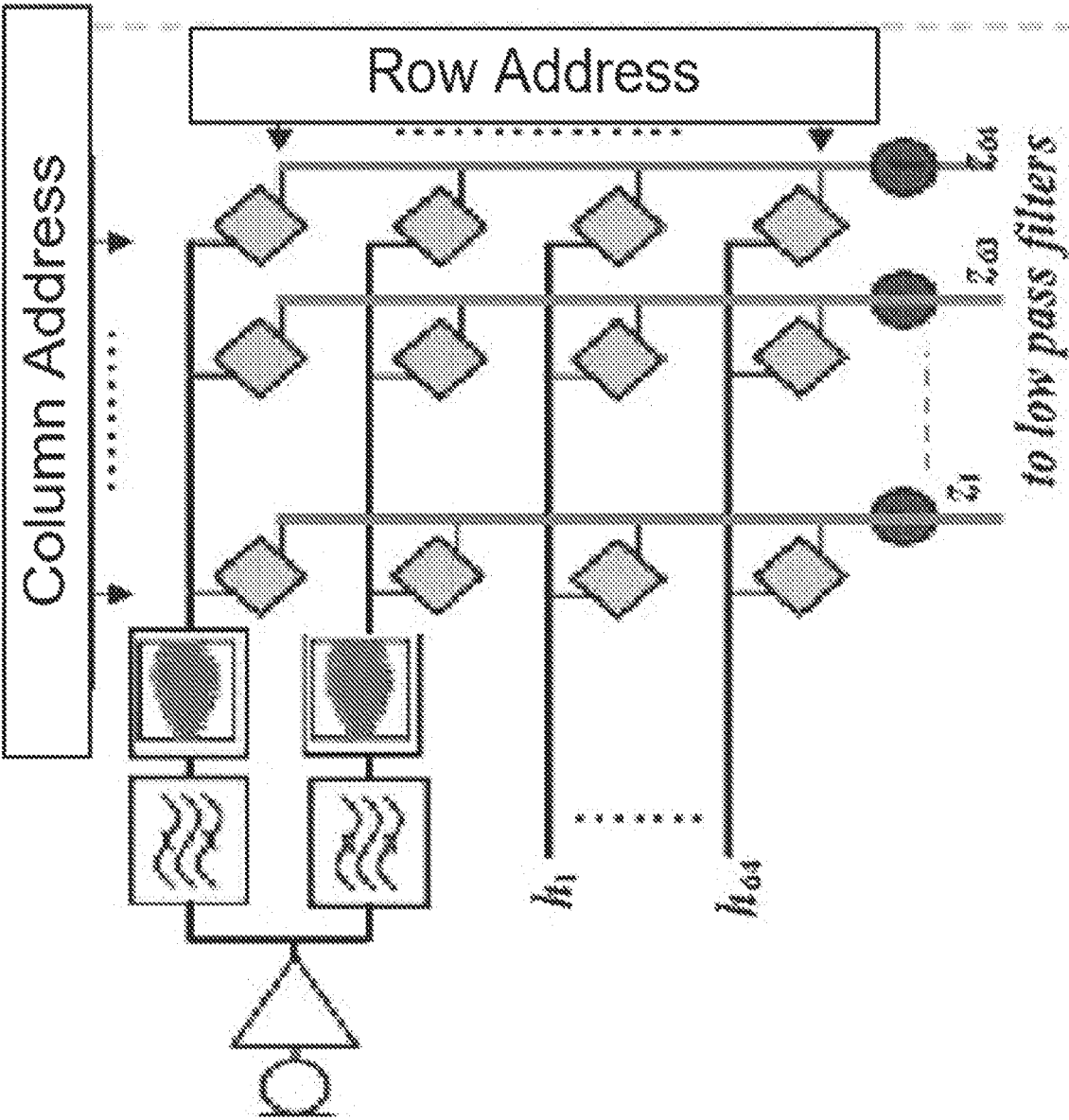


Fig. 5A

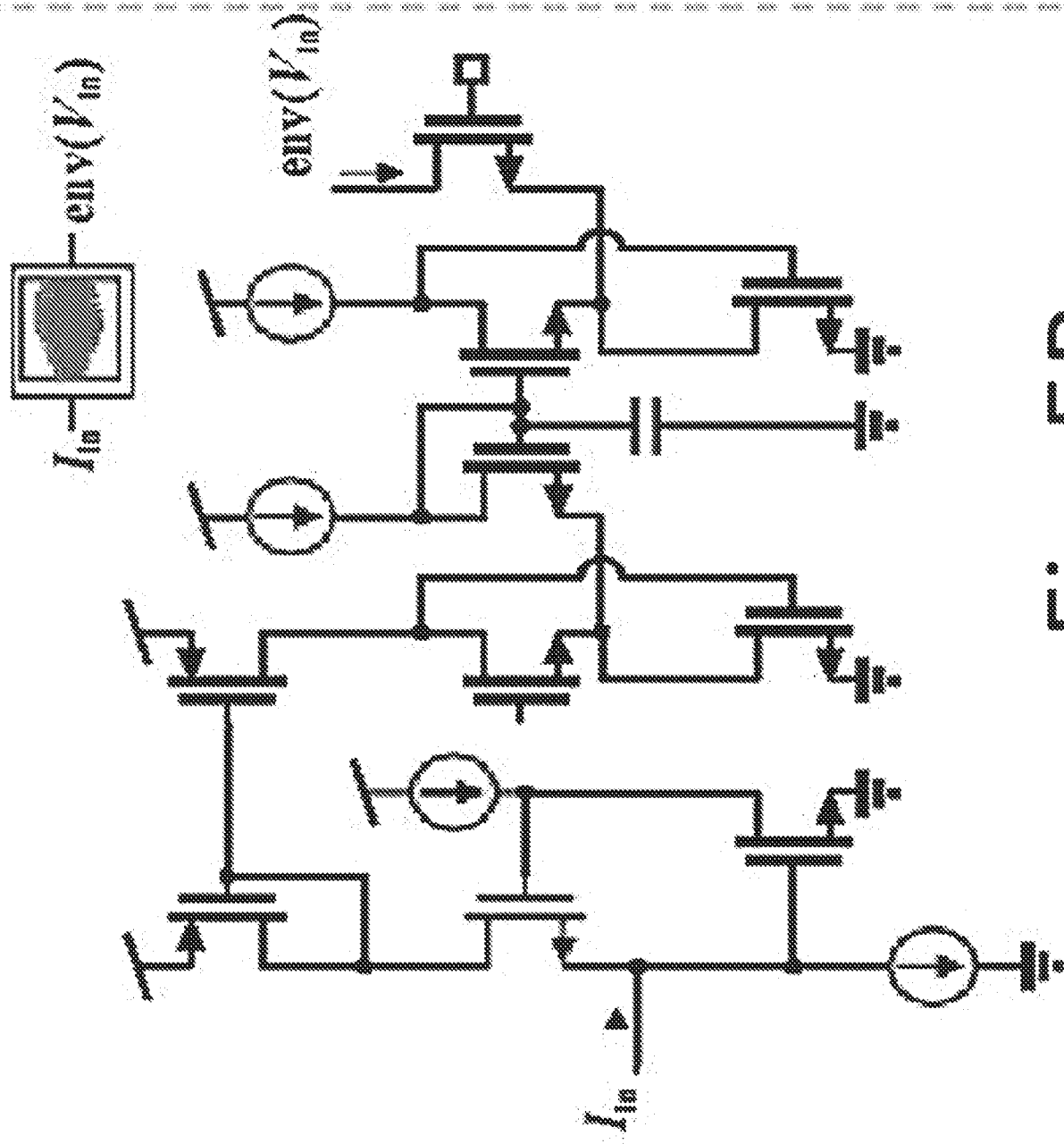


Fig. 5B

Fig. 5C

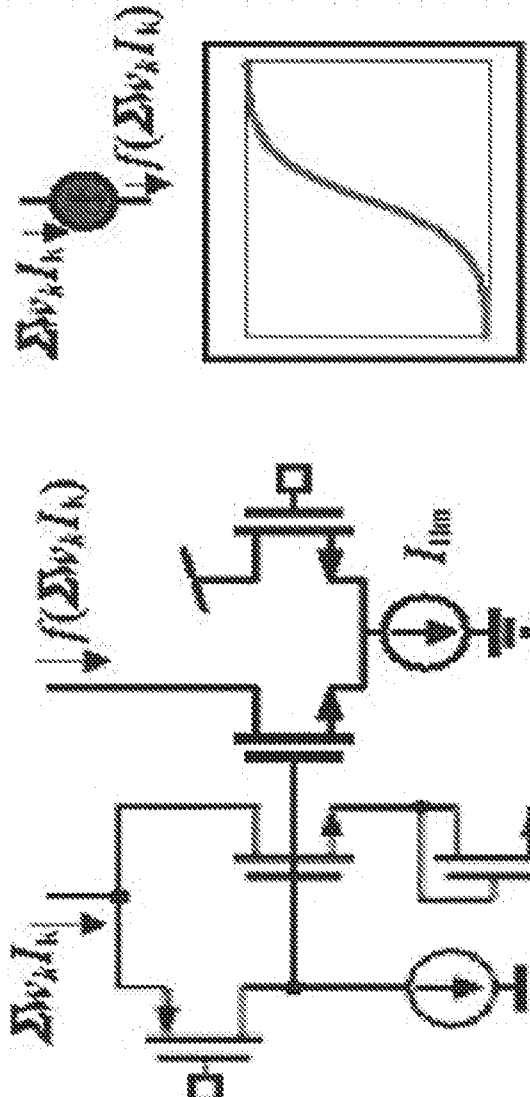
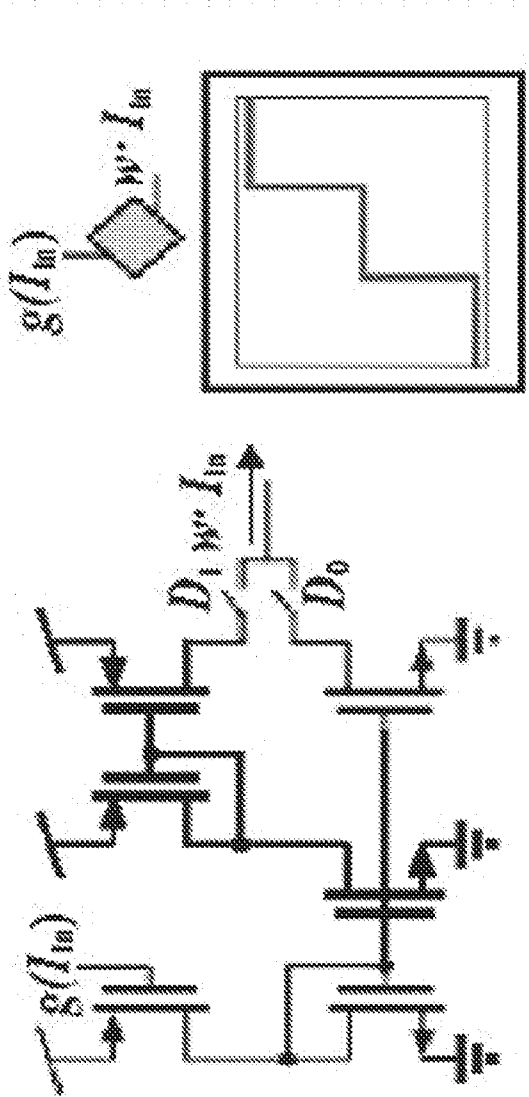


Fig. 5D

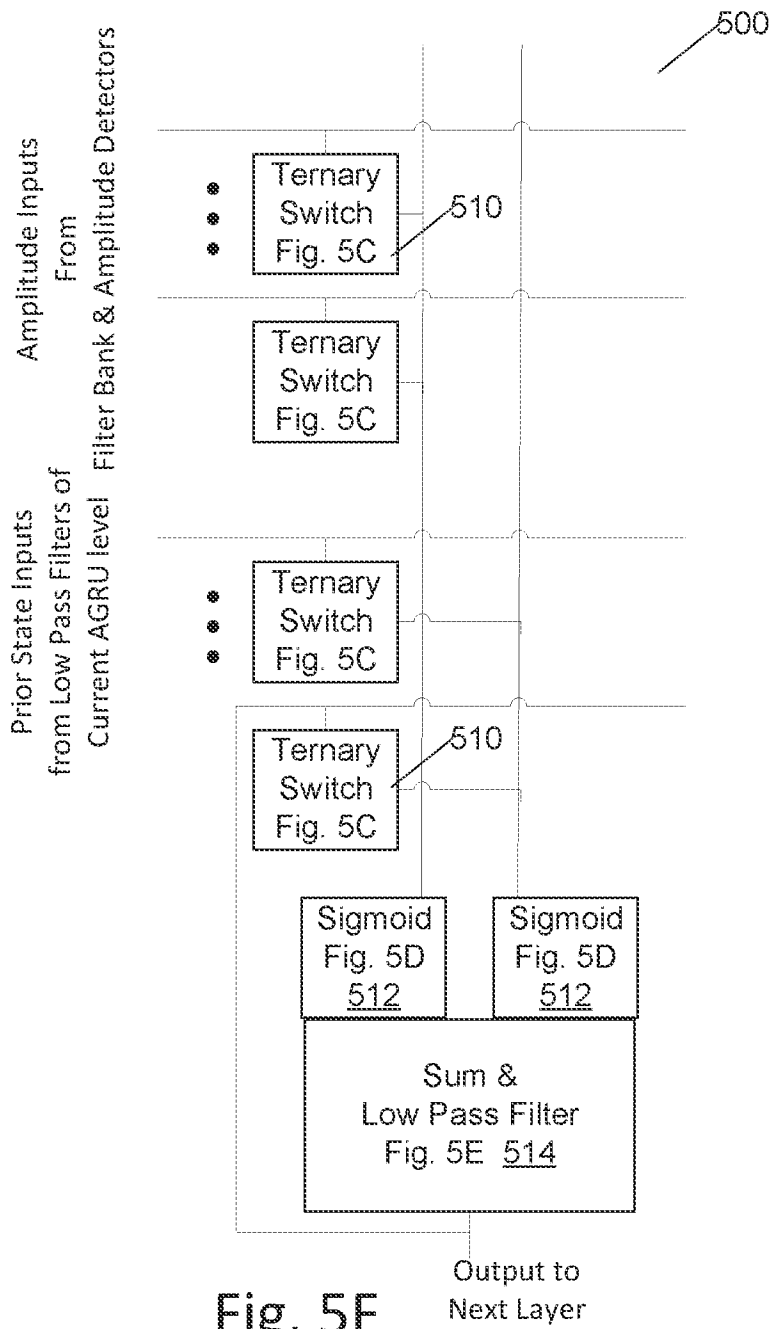


Fig. 5F

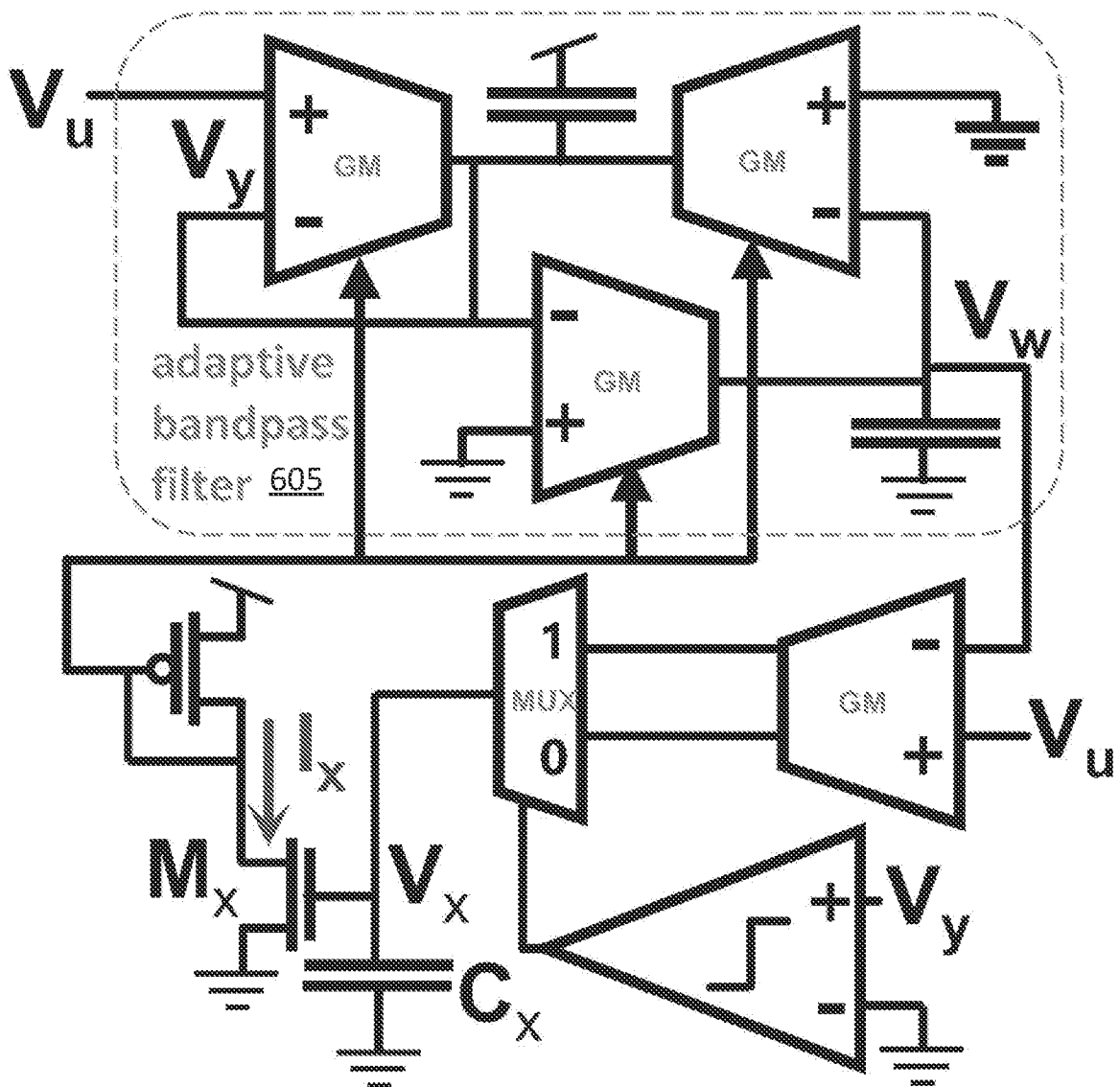
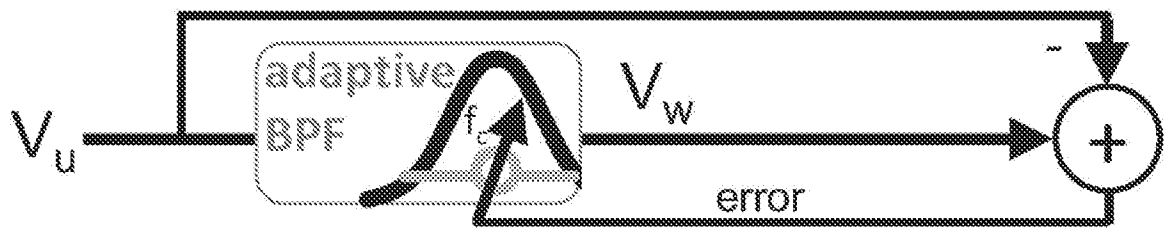


Fig. 6B

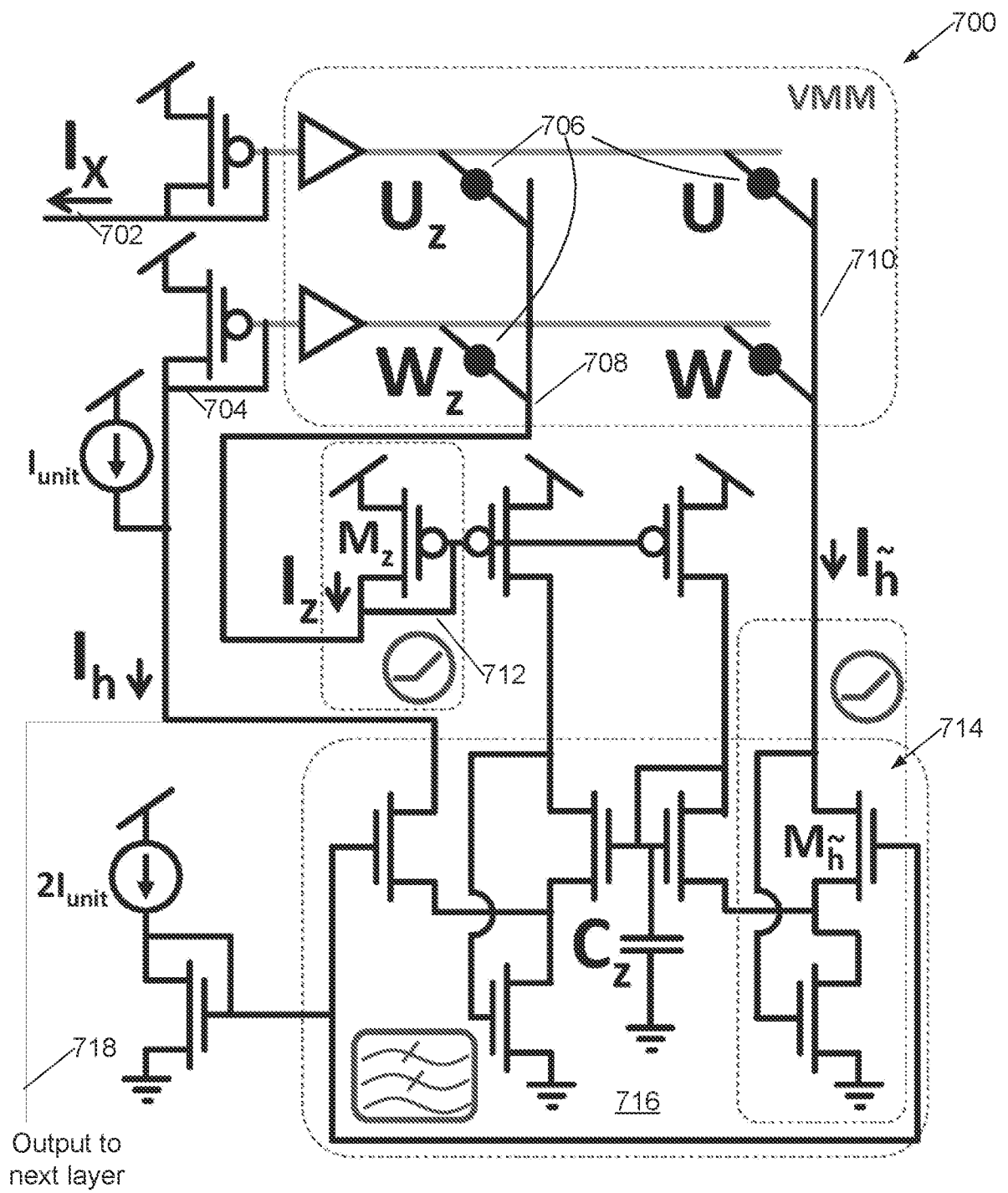


Fig. 7

INTERNATIONAL SEARCH REPORT

International application No.

PCT/US22/26416

A. CLASSIFICATION OF SUBJECT MATTER

IPC - INV. G06N 3/06; G06F 17/16; G06N 3/02; G06N 3/04; G10L 21/0208 (2022.01)

ADD.

CPC - INV. G06N 3/06; G06F 17/16; G06N 3/02; G06N 3/04; G10L 21/0208

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)

See Search History document

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

See Search History document

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

See Search History document

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y --- A	SAK, H ET AL.. "LONG SHORT-TERM MEMORY RECURRENT NEURAL NETWORK ARCHITECTURES FOR LARGE SCALE ACOUSTIC MODELING". INTERSPEECH. . 2014; pages 338-342. [Retrieved on 8th August 2022] [online] <url: https://www.aminer.cn/pub/555044af45ce0a409eb50004/long-short-term-memory-recurrent-neural-network-architectures-for-large-scale-acoustic >; abstract, figure 2d, page 338, column 2, paragraph 4-page 339, column 1, paragraph 1, page 339, column 1, paragraph 3-column 2, paragraph 1, page 340, column 1, paragraph 3	1-4, 9 --- 5-8, 10-17
Y --- A	US 2020/0065660 A1 (SILICON STORAGE TECHNOLOGY, INC.) 27 February 2020; paragraphs [0088], [0089], [0108], [0109], [0112], [0113]	1-4, 9 --- 5-8, 10-17
Y	WO 2016/053822 A1 (MICROSOFT TECHNOLOGY LICENSING, LLC) 07 April 2016; paragraph [0122], claim 1	2-4, 9
Y	US 5,323,467 A (HERMES, D) 21 June 1994; figure 3, column 4, line 48-column 5, line 15	3, 4
Y	US 7,937,155 B1 (VOELKEL, A) 03 May 2011; column 5, lines 22-24	4
Y --- A	BEEGUM, N. "Speaker Identification with Whispered Speech". International Journal of Advanced Research in Computer and Communication Engineering. Online. June 2016; Vol. 5, Issue 6. [Retrieved on 8th August 2022] [online] <URL: https://ijarcce.com/upload/2016/june-16/IJARCCE%20122.pdf >; page 565, column 2, paragraph 3-page 566, column 1, paragraph 1	9 --- 10-12

☒ 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

"D" document cited by the applicant in the international application

"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

10 August 2022 (10.08.2022)

Date of mailing of the international search report

SEP 08 2022

Name and mailing address of the ISA/US

Mail Stop PCT, Attn: ISA/US, Commissioner for Patents
P.O. Box 1450, Alexandria, Virginia 22313-1450
Facsimile No. 571-273-8300

Authorized officer

Shane Thomas

Telephone No. PCT Helpdesk: 571-272-4300

INTERNATIONAL SEARCH REPORT

International application No.

PCT/US22/26416

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2016/0322055 A1 (GOOGLE INC.) 03 November 2016; the entire document	1-17
A	US 2014/0046885 A1 (MAJUMDAR, S ET AL.) 13 February 2014; the entire document	1-17
A	US 10,977,547 B2 (GOOGLE LLC.) 13 April 2021; the entire document	1-17
A	KIHAS, D ET AL.. "ADAPTIVE FILTERING BASED ON RECURRENT NEURAL NETWORKS" JOURNAL OF AUTOMATIC CONTROL, UNIVERSITY OF BELGRADE, VOL. 13(1). Web. 2003; pages 13-24. [Retrieved on 9th August 2022] [online] <URL:http://www.doiserbia.nb.rs/img/doi/1450-9903/2003/1450-99030301013K.pdf>; abstract, page 14, column 2, paragraph 3	1-17