Journal of Applied Geophysics 181 (2020) 104160

Contents lists available at ScienceDirect

Journal of Applied Geophysics

journal homepage: www.elsevier.com/locate/jappgeo

Prestack time migration velocity analysis using recurrent neural networks

Deborah Pereg [a,*], Israel Cohen [a], Anthony A. Vassiliou [b], Rod Stromberg [b]

[a] Technion – Israel Institute of Technology, Israel
[b] GeoEnergy, Inc., United States

ARTICLE INFO

Article history:
Received 3 January 2020
Received in revised form 24 May 2020
Accepted 11 August 2020
Available online 16 August 2020

Keywords:
Prestack time migration
Seismic inversion
Seismic signal
Recurrent neural net

ABSTRACT

We present a new efficient method to perform prestack time migration velocity analysis (MVA) of seismic data based on recurrent neural nets (RNNs). We assume that there exits a mapping from each local data subvolume, which we term an analysis volume, to each single velocity point value in space. Under this hypothesis, as RNNs are capable of learning structural information both in time and in space, we propose designing a net that can learn this mapping, and exploit it to produce the entire root-mean-square (rms) velocity field. The performance of the method is evaluated via real data experimental results in comparison to existing technique for MVA. We present different aspects of the system's behavior with various training and testing scenarios and discuss potential advantages and disadvantages. We also recast the same RNN to learn and automate migration velocity picking from constant velocity migration (CVM) panels and compare real data results for this alternative. Our approach is extremely efficient in terms of computational complexity and running time, and therefore can be potentially applied to large volumes of three-dimensional (3D) seismic data, and significantly reduce work load. An extension to four-dimensional (4D) data is also possible.

1. Introduction

A fundamental task in exploration seismology is the visualization of earth internal structure. Seismic imaging enables us to *see into* the subsurface earth layers, and examine specific geological structures (e.g. layers, channels, traps and faults), which in turn facilitates exploration of mineral deposits and energy sources (Berkhout, 1986), harvesting of geological information for engineering, geothermal energy surveys, risk assessment of tsunamis, and more.

One of the major challenges in seismic imaging is to improve image resolution. Resolution limits are inherently determined by the wave length, thus, by diffraction. In seismic data time processing, there are three basic stages executed in a varying order: migration, stacking and deconvolution (Biondi, 2006; Sheriff and Geldart, 1983; Öz Yilmaz et al., 2001). Seismic migration focuses the data, collapses diffraction curves, and positions the dipping reflectors in their true locations.

Accurate migration and interpretation of seismic data requires knowledge of the velocity of the propagating waves at all points along the reflection paths. In order to perform migration correctly, and produce the image at depth, reliable estimation of the velocity of the propagating waves is essential. In fact, estimating 3D velocity is considered as one of the most important problems in exploration geophysics (Sava and Biondi, 2004). Furthermore, despite rapid technological progress, some stages in migration are still currently done manually, which makes the entire process very slow, especially when dealing with huge datasets.

Generally speaking, standard prestack time migration velocity analysis (MVA) (Biondi, 2006; Sheriff and Geldart, 1983; Öz Yilmaz et al., 2001) is an iterative process of two main stages: (1) A decimated dataset is imaged by prestack migration. (2) Velocity model is updated based on some cost function, such as semblance along the offset axis. The most efficient prestack time MVA is accomplished by employing constant migration velocity volumes (CVM) (Öz Yilmaz et al., 2001). MVA is considered as one of the most significant and time-consuming stages in geophysical data processing. Errors in this stage result in critical deterioration of the produced subsurface image, especially at depth.

Nowadays seismic acquisition concentrates more on 3D data, and the amount of data is rapidly growing. Unfortunately, prestack time migration velocity analysis and picking methods are time consuming. Recently, deep learning (DL) methods achieved remarkable results in many tasks in signal processing and image processing. In exploration seismology, learning methods were successfully employed to various tasks. Calderón-Maciás et al. (1998) proposed the use of feedforward neural network (FNN) for normal moveout (NMO) velocity estimation. Artificial neural networks (ANNs) were also employed to automate first arrival picking (see e.g. Murat and Rudman (1992) McCormack and Rock (1993); Kahrizi and Hashemi (2014); Mężyk and Malinowski

* Corresponding author.
E-mail address: deborahp@campus.technion.ac.il (D. Pereg).

https://doi.org/10.1016/j.jappgeo.2020.104160
0926-9851/

(2019)). Hollander et al. (2018) proposed the use of convolutional neural nets (CNNs) to this task as well. These techniques can be separated into two main categories: Computing an attribute of the data and then feeding it into a net, which is then used for classification; as opposed to feeding patches of the data directly into a neural net that is in charge of both extracting the relevant features from the data, and then classifying it accordingly. Moreover, in Pereg et al. (2020) we propose the use of recurrent neural nets (RNNs) for deconvolution of seismic data, achieving impressive results.

Attempts have been made to apply DL methods to seismic velocity estimation as well. Fish and Kusuma (1994) explored velocity picking automation by using a neural net trying to imitate the human process, requiring picked velocities to be associated with semblance peaks. RNNs were also successfully used to estimate stacking velocity directly from seismic data for NMO correction by Biswas et al. (2018). Furthermore, Li et al. (2019) proposed a method to build a seismic velocity model from seismic data via DNNs.

Application of neural networks to seismic interpretation was also explored. For example, Dorrington and Link (2004) suggested to incorporate a neural net in a genetic algorithm in an attempt to find optimal seismic attributes for well-log prediction; Araya-Polo et al. (2017) and Zhang et al. (2014) proposed to automate fault detection from seismic data before migration by using CNNs; Wang et al. (2018) proposed the use of CNNs for detection of salt dome boundaries. In addition, Kumar and Sain (2018) and Kumar et al. (2019a, 2019b, 2019c) proposed automated techniques for enhancing interpretation of faults, magmatic sills, fluid migrations and buried volcanic systems from seismic data. Also, Haris et al. (2017, 2018) propose to employ neural nets to seismic data, for learning of sonic properties, and for pore pressure prediction.

In this paper, our primary purpose is to develop an automated fast and efficient technique for prestack seismic time migration velocity analysis. To this end, we suggest that there exists a mapping from each subvolume of seismic data to each single velocity point value in space. Moreover, we postulate that a neural net can learn this mapping, using a relatively small subset of the data for training. We presume that RNN is the most equipped for this task, since it incorporates information of both spatial and temporal relations in the data. We assume that the mapping from each analysis volume to a velocity point value is the same for the sake of simplicity. In other words, the data is assumed to be stationary. In practice this presumption is not completely true, but, as will be described, this simplification is essential for analyzing the major concepts of this line of research and for evaluation of its competence. In addition, we also propose to adjust the application of an RNN to migration velocity picking from CVM panels, and compare real data results of the two suggested approaches.

The paper is organized as follows. In Section 2, we provide background for Kirchhoff migration, MVA and RNNs. In Section 3, we describe the proposed method for MVA using RNN. Section 4 presents experimental real data results. In Section 5, we apply the proposed method to automated migration velocity picking using CVM panels. Lastly in Section 6 we summarize and discuss further research directions.

2. Background and related work

2.1. Seismic migration

In seismic imaging (Biondi, 2006; Sheriff and Geldart, 1983), migration is a process designed to relocate the reflectors to their true location. Migration is applied to correct the position of the reflectors and focus the data to produce a subsurface image. In other words, given the vertically plotted points, migration is the procedure of finding the true reflecting surfaces (Schneider, 1978; Hagedoorn, 1954).

2.2. Kirchhoff migration

Consider a source-receiver with zero offset and a dipping reflector. The recorded data before migration is *plotted vertically* below the source-receiver. However, the reflection could have originated from anywhere along the surface of points with the same reflection time, a semicircle - in the case of constant velocity. If we plot a few semicircles for a few recorded traces we can see that they constructively interfere to create a dipping reflector in the correct location. Migration algorithms of this type are often referred to as spraying methods.

In the case of non-zero source-receiver offset, a reflection time $2T$ observed at the receiver corresponds to a reflection from a surface that is tangential at some point to the "surface of equal reflection times" (Hagedoorn, 1954). Namely, each point on the surface of equal reflection times is an intersection of wavefronts surfaces at times $T + t$ from the source, and wavefront surfaces at times $T - t$ from the receiver, for all values of t. Hence, to an observer at the receiver, the true location of the reflector on this surface is ambiguous. It is worth mentioning that in his work Hagedoorn (1954) refers to the surfaces of equal reflection times as "the surfaces of maximum observable concavity", because, due to practical considerations, when each point results in a vertically plotted point, a reflecting surface cannot be more concave then a surface of equal reflection times.

Let us denote the vertical depth axis as z, where $z = 0$ is the ground surface, and (x,y) are the location coordinates along the horizontal directions perpendicular to z. Each point on a reflector produces a *diffraction curve* in the gather image (Hagedoorn, 1954). As known, in a constant velocity case, the diffraction curve is a hyperbola. The correct location of the point would be at the apex of the hyperbola. In other words, suppose a point reflector at location (x_0,z_0), where (x,z) are the location coordinates along a 2D section of the ground. When the point source "explodes" at $t = 0$ the data recorded at a geophone on ground, i.e., at $z = 0$, as a function of a location x at traveltime t is an impulsive signal (a reflection) along a trajectory $t^2 = \frac{(x-x_0)^2+z_0^2}{V^2}$, where V is a constant acoustic wave velocity. According to the principle of Huygens, a line, which can be treated as a sequence of points, produces a superposition of hyperbolas. In other words, a reflector is treated as a sequence of closely spaced diffracting points. Eventually, the recorded data is a superposition of all hyperbolic arrivals. If we were to perform migration manually, we would have plotted the diffraction curves for each unmigrated point and slide them along the unmigrated image until the best tangential fitted. In the migrated data, the reflector is positioned at the apex of the diffraction curve tangent to the wavefront passing through the apex point. Namely, the diffraction curve collapses to its apex.

Practically, in order to perform migration, we solve the wave equation. Different types of migration methods solve the wave equation in different ways. In many cases, prestack migration is performed using Kirchoff method (Schneider, 1978; Biondi, 2006). Assume the source is located at $\mathbf{s} = (x_s,y_s)$, and the receiver is located at $\mathbf{g} = (x_g,y_g)$. The data coordinates expressed in terms of source-receiver coordinates are called *field coordinates*, as opposed to *midpoint-offset coordinates* $(\mathbf{m},\mathbf{h})$ defined as:

$$\left(\mathbf{m} = \frac{\mathbf{g}+\mathbf{s}}{2}, \mathbf{h} = \frac{\mathbf{g}-\mathbf{s}}{2}\right).$$

Starting with the scalar wave equation

$$\nabla^2 P = \frac{1}{v^2(x,y,z)} \frac{\partial^2 P}{\partial t^2}, \tag{1}$$

where $P(x,y,z;t)$ is the pressure wavefield propagating in a medium with velocity $v(x,y,z)$. In our case, we have a homogenous wave equation with inhomogeneous boundary conditions, since there are no real sources in the subsurface image, only reflectors and scatterers. Namely,

$P(x_0,y_0,z_0;t_0)$ is the observed seismic data, and $U(x,y,z) \triangleq P(x,y,z;t=0)$ is the migrated image.

The general form of Kirchhoff migration is given by

$$U(\mathbf{r}) = \int_{\Omega_r} W(\mathbf{r},\mathbf{m},\mathbf{h}) D(t = t_D(\mathbf{r},\mathbf{m},\mathbf{h}),\mathbf{m},\mathbf{h}) d\mathbf{m} d\mathbf{h}, \tag{2}$$

where $\mathbf{r} = (x,y,z)$ is a location in the 3D space, $D(t = t_D,\mathbf{m},\mathbf{h})$ are the data values recorded at time $t_D(\mathbf{r},\mathbf{m},\mathbf{h})$ and weighted by a factor $W(\mathbf{r},\mathbf{m},\mathbf{h})$. The total time delay $t_D(\mathbf{r},\mathbf{m},\mathbf{h})$ is the time accumulated as the reflections propagate from the source to the image point and back to the surface to be recorded at the receiver position. The integration domain Ω_r is the migration aperture. The summation domain does not include the whole input space; it is bounded to a region centered around the location $\mathbf{r}$ in the midpoint $\mathbf{m}$ plane. The summation domain affects the dip limits and the computational cost of the migration procedure. Deriving the weights $W(\mathbf{r},\mathbf{m},\mathbf{h})$ is a complicated task. We refer the reader to Cohen and Bleistein (1979); Beylkin (1985); Schleicher et al. (1993) for further details.

In practice, prestack data are recorded on a discrete set of points. Therefore, the integral in (2) is approximated with a finite sum

$$U(\mathbf{r}) \approx \sum_{i \in \Omega_r} W(\mathbf{r},\mathbf{m}_i,\mathbf{h}_i) D(t = t_D(\mathbf{r},\mathbf{m}_i,\mathbf{h}_i),\mathbf{m}_i,\mathbf{h}_i). \tag{3}$$

There are two types of algorithms implementing the summation: Gathering methods and spraying methods. Gathering methods assemble the contributions of all input traces within the migration aperture for each image point, while spraying methods spray the data of the input traces over all image points within the migration aperture.

Assuming constant velocity, the time delays define a family of summation surfaces

$$t_D(x,y,z,\mathbf{m},\mathbf{h}) = \frac{\sqrt{z^2 + |(x,y) - \mathbf{m} + \mathbf{h}|^2}}{V} + \frac{\sqrt{z^2 + |(x,y) - \mathbf{m} - \mathbf{h}|^2}}{V}. \tag{4}$$

When the velocity is not constant, the summation surfaces have more complex shapes, computed numerically through a complex velocity function. All methods solve the Eikonal equation, that is, an approximation of the wave equation (Bleinstein, 1984).

When velocity changes slowly in the horizontal direction, we can approximate the time delay functions in (4) using root-means-square (rms) velocity V_{rms} instead of a constant velocity. The average velocity V_{rms} related to a specific raypath is defined as

$$V_{\text{rms}}^2 = \frac{\int_0^t v^2(t) dt}{\int_0^t dt}, \tag{5}$$

where t is the time required for the wave to traverse the path. The rms velocity V_{rms} is tied to the interval velocity function $\overline{v}(\tau)$, i.e., the average velocity over some travel path, by the Dix formula (Dix, 1955). In a layered medium,

$$\overline{v}^2(\tau_i) = \frac{\tau_i V_{\text{rms}}^2(\tau_i) - \tau_{i-1} V_{\text{rms}}^2(\tau_{i-1})}{\Delta\tau_i}, \tag{6}$$

where $\Delta\tau_i$ is the time thickness of the ith layer, $\overline{v}(\tau_i)$ is the interval velocity within the ith layer, τ_i and τ_{i-1} are the corresponding two-way zero-offset times of the layer boundaries, and $V_{\text{rms}}(\tau_i)$ and $V_{\text{rms}}(\tau_{i-1})$ are the corresponding rms velocities. Conversely,

$$V_{\text{rms}}^2(\tau_N) = \frac{\sum_{i=1}^N \overline{v}^2(\tau_i)\Delta\tau_i}{\sum_{i=1}^N \Delta\tau_i}, \tag{7}$$

where $\tau_N = \sum_i \tau_i$ is the total two-way traveltime to the bottom of the N'th layer. It is worth mentioning that the names time migration and depth migration do not refer to the vertical axis of the image. Time migrated images are computed by analytically computing the time delay function using the average velocity estimated at each image location. Namely,

$$t_D(x,y,z,\mathbf{m},\mathbf{h}) = \frac{\sqrt{z^2 + |(x,y) - \mathbf{m} + \mathbf{h}|^2}}{V_{\text{rms}}(x,y,z)} + \frac{\sqrt{z^2 + |(x,y) - \mathbf{m} - \mathbf{h}|^2}}{V_{\text{rms}}(x,y,z)}. \tag{8}$$

On the other hand, depth migrated images are computed using the interval velocity estimated at every point in the subsurface.

2.3. Migration velocity analysis

Clearly, accurate interpretation of seismic data requires knowledge of the velocity at all points along the reflection paths. In order to perform migration correctly, and produce the image at depth, reliable estimation of the velocity of the propagating waves is needed.

The estimation of velocity from a given seismic data is considered as an ill-posed inverse problem, because it is not clear whether the data holds all the necessary information to compute a velocity function that varies with depth and along the horizontal directions. As a result, we assume having some prior knowledge, that can be sufficient to define and solve a constrained problem.

As we stated before, migration focuses the data and assigns reflectors to their true locations. Migration velocity analysis (MVA) methods are velocity-estimation methods that use migrated data to extract kinematic information, which is then used again to migrate the data or perform depth migration. Essentially, MVA is an iterative process of two stages: (1) Data is imaged by prestack migration; (2) Velocity function is updated based on the migration results. The accuracy of the velocity function is determined by measuring the focusing of reflections in the data domain or in the image domain. The most common criterion is the coherence of the data in common-midpoint (CMP) gathers along the offset domain after application of normal moveout (NMO) correction. Alternatively, starting with unmigrated data we can generate CVM panels and extract time slices from the CVMs. For visualization, we form a super image by placing the CVMs one on top of the other to yield a volume of migrated data of midpoint, time, and migration velocity axes. Then, for each time slab and each midpoint we pick the maximum event associated with a velocity value, in the same manner as for maximum semblance picking (Öz Yilmaz et al., 2001). This is commonly done by picking structurally consistent horizon strands of the highest amplitude along the midpoint direction from the CVM panels. The corresponding velocities of all time strands form the rms velocity field. This process can be repeated to improve the velocity estimation results.

2.4. Recurrent neural net

In classic feed forward networks (FNNs) information travels in one direction, that is - from input to output. In contrast, in a recurrent neural network (RNN) the nodes of the graph are connected by feedback connections, in addition to the feedforward connections (Hochreiter and Schmidhuber, 1997; Géron, 2017), which means the signal also travels backwards. In other words, at a current state, the output depends on current inputs, and on outputs at previous states. In a sense, we can say that the network is able to make decisions that are based also on the memory of its previous decisions.

Mathematically speaking, let us assume an input sequence $\mathbf{x} = [\mathbf{x}_0, \mathbf{x}_1,\ldots,\mathbf{x}_{L_t-1}]$, and a corresponding output sequence $\mathbf{y} = [\mathbf{y}_0,\mathbf{y}_1,\ldots,\mathbf{y}_{L_t-1}]$. In supervised learning, the RNN forms a map $f : \mathbf{x} \to \mathbf{y}$, from the input data to its labels or to a predicted function. The output of the net at time step t is defined as

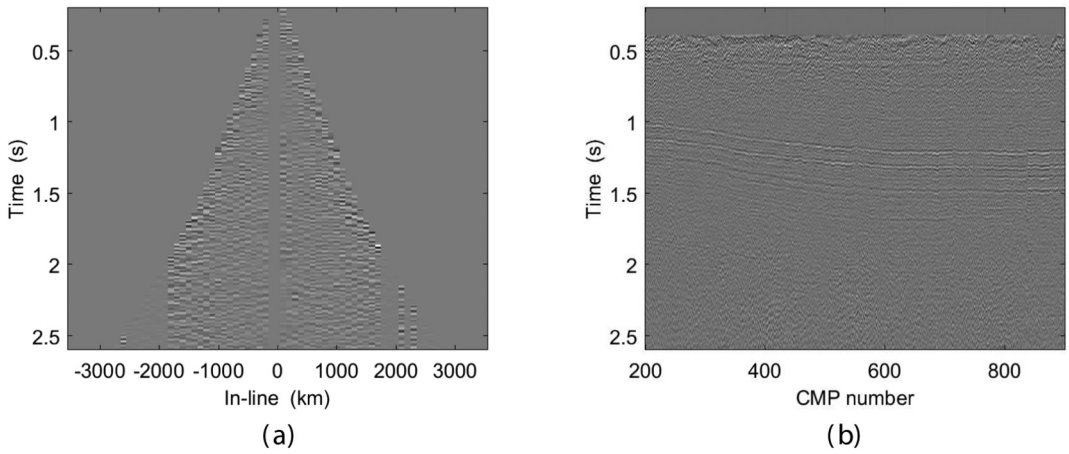

Fig. 1. Real seismic data: (a) CMP gather with 71 offsets; (b) common offset gather with 700 CMP traces.

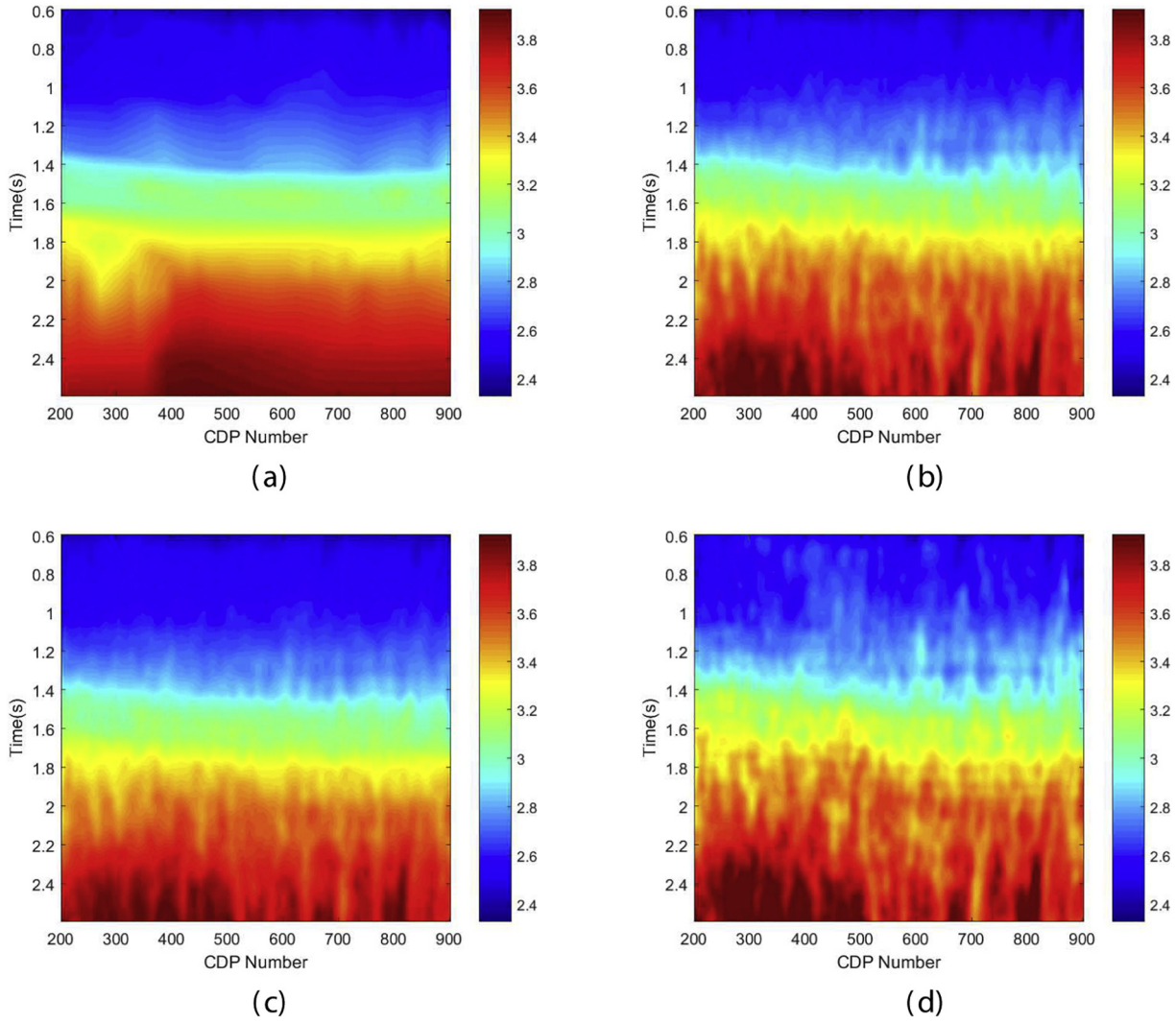

Fig. 2. Migration velocities: (a) picked velocities used as ground truth; (b) predicted single channel velocities, $N_m = 1$, with 25% training; (c) predicted single channel velocities, $N_m = 1$ with 50% training; (d) predicted multichannel velocities, $N_m = 3$, with 25% training.

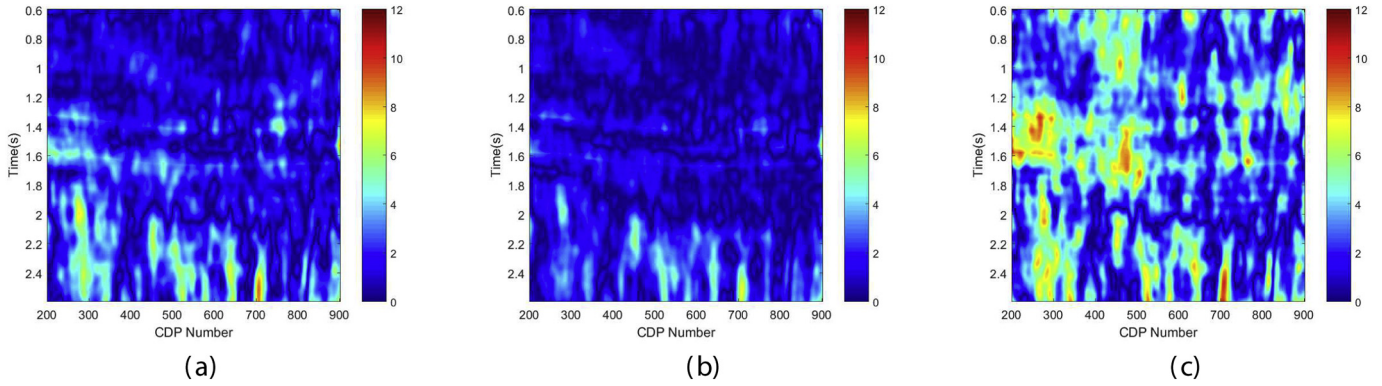

Fig. 3. Percentage error: (a) predicted single-channel velocities, $N_m = 1$ with 25% training; (b) predicted single channel velocities, $N_m = 1$, with 50% training; (c) predicted multichannel velocities, $N_m = 3$, with 25% training.

$$\mathbf{y}_t = \sigma\left(\mathbf{W}_{xy}^T \mathbf{x}_t + \mathbf{W}_{yy}^T \mathbf{y}_{t-1} + \mathbf{b}\right), \tag{9}$$

where σ is an activation function, $\mathbf{W}_{xy}$ and $\mathbf{W}_{yy}$ are weight matrices and $\mathbf{b}$ is the bias vector. We assume that at time step $t = 0$ previous outputs are zero. The function σ can be one of the known activation functions, for example: Sigmoid, rectified linear units (ReLU) and hyperbolic tangent. In our implementation we use the ReLU activation function, $\mathrm{ReLU}(z) = \max\{0, z\}$. Only one layer of recurrent neurons is employed for this task, but multiple layers can be employed to form a deep neural network.

Let us denote n_i as the number of inputs at time step t, namely $\mathbf{x}_t$ is of size $n_\mathrm{i} \times 1$, and n_n as a predetermined number of neurons in an RNN cell. Clearly $\mathbf{W}_{xy}$ is of size $n_\mathrm{i} \times n_\mathrm{n}$, and $\mathbf{W}_{yy}$ is of size $n_\mathrm{n} \times n_\mathrm{n}$. Practically, during training, at each iteration the weights are updated for multiple instance sequences of data referred to as a mini-batch. Suppose we have m instances in a mini-batch, then the output can be simply written as

$$\mathbf{Y}_t = \sigma\left(\mathbf{X}_t \mathbf{W}_{xy}^T + \mathbf{Y}_{t-1} \mathbf{W}_{yy}^T + \mathbf{b}\right), \tag{10}$$

where $\mathbf{Y}_t$ is of size $m \times n_\mathrm{n}$, and $\mathbf{X}_t$ is of size $m \times n_\mathrm{i}$.

Clearly, it follows that the output at time step t depends on outputs of previous time steps. Hence, as mentioned, it is said that the RNN has memory. Essentially, the RNN absorbs a time-series input and produces a time-series output. Due to this property RNNs are often used in applications that require processing of time related signals, or to predict future outcomes.

At this stage, the output $\mathbf{y}_t$ of each recurrent neuron cell is of size $n_\mathrm{n} \times 1$. To fit the size of the output to our purpose, we wrap the cell with a fully connected layer with the desired final output $\mathbf{z}_t \in \mathbb{R}^{1 \times M}$, such that

$$\mathbf{z}_t = \mathrm{FC}(\mathbf{y}_t). \tag{11}$$

Let us denote by $\mathbf{Z} \in \mathbb{R}^{L_t \times M}$ the matrix of predicted outputs, that is the concatenation of the output vectors $\mathbf{z}_t$, $t = 0, 1, \ldots, L_t - 1$ as columns, and define the weights matrices and the bias as $\theta = \{\mathbf{W}_{xy}, \mathbf{W}_{yy}, \mathbf{b}\}$. In our application, the loss function is the mean squared error of the predicted output and the expected output. Suppose $\mathbf{Z}_i$ and $\widehat{\mathbf{Z}}_i$ are the expected output during training and current system's output, respectively, for input sequence $\mathbf{x}_i$, and denote the error matrix as $\mathbf{E}_i = \mathbf{Z}_i - \widehat{\mathbf{Z}}_i$.

$$J(\theta) = \frac{1}{mML_t} \sum_{i=1}^{m} tr\left(\mathbf{E}_i^T \mathbf{E}_i\right), \tag{12}$$

where superscript ^{T} denotes the transpose of a vector or a matrix, and $tr(\cdot)$ denotes the trace of a square matrix. During training, after each forward pass of a mini-batch, the gradient of the loss function is computed using back-propagation and the weights are updated using Adam-optimization, with a defined learning rate value (Chen, 2016).

3. Seismic MVA using RNN

Let us denote $\mathbf{S} \in \mathbb{R}^{L_s \times J_\mathrm{m} \times J_\mathrm{o}}$ as an observed seismic data with $J_\mathrm{m} \times J_\mathrm{o}$ traces of L_s time samples along the midpoint and offset directions. In addition, we denote the ground truth prestack time migration velocity field as $\mathbf{V} \in \mathbb{R}^{L_s \times J_\mathrm{m}}$, and the predicted velocity field as $\widehat{\mathbf{V}} \in \mathrm{R}^{L_s \times J_\mathrm{m}}$. Note that we do not assume any specific prior regarding the structure of the data.

Definition 1 (Analysis volume): We define an *analysis volume* as a 3D volume of size $L_t \times N_\mathrm{m} \times N_\mathrm{o}$ enclosing L_t time (depth) samples of $N_\mathrm{m} \times N_\mathrm{o}$ consecutive traces of the observed seismic data $\mathbf{S}$. In other words, we define a subvolume of the data, that consists of a time window of L_t samples, of N_o offset channels corresponding to N_m CMPs. Assume $\{n_{\mathrm{m,L}}, n_{\mathrm{m,R}} \in \mathbb{N} : n_{\mathrm{m,L}} + n_{\mathrm{m,R}} = N_\mathrm{m} - 1\}$, then the analysis volume associated with a point at time (depth) k at CMP index m is

$$\mathbf{A}_{k,m} = \begin{pmatrix} \widetilde{\mathbf{S}}_{k-L_t+1, m-n_{\mathrm{m,L}}} & \cdots & \widetilde{\mathbf{S}}_{k-L_t+1, m+n_{\mathrm{m,R}}} \\ \vdots & \ddots & \vdots \\ \widetilde{\mathbf{S}}_{k, m-n_{\mathrm{m,L}}} & \cdots & \widetilde{\mathbf{S}}_{k, m+n_{\mathrm{m,R}}} \end{pmatrix}, \tag{13}$$

where we define

$$\widetilde{\mathbf{S}}_{k,m} = \left[\mathbf{S}_{k,m,n_{\mathrm{ZO}}-n_{\mathrm{o,L}}}, \ldots, \mathbf{S}_{k,m,n_{\mathrm{ZO}}+n_{\mathrm{o,R}}}\right]$$

as a section of N_o offsets of the data, $\{n_{\mathrm{o,L}}, n_{\mathrm{o,R}} \in \mathbb{N} : n_{\mathrm{o,L}} + n_{\mathrm{o,R}} = N_\mathrm{o} - 1\}$, and n_{ZO} is the zero-offset index. Notice that each analysis volume must be normalized to have zero mean and unit variance, to ensure that all inputs belong to the same probability distribution.

An analysis volume $\mathbf{A}_{k,m}$ is associated with a velocity field point $\mathbf{V}_{k,m}$. In order to find a velocity value $\mathbf{V}_{k,m}$, we define the input to the RNN as

$$\mathbf{x} = \mathbf{A}_{k,m}. \tag{14}$$

Accordingly, each time step input is a data section of $N_\mathrm{m} \times N_\mathrm{o}$ neighboring data points at the corresponding time (depth). Namely, for this application $n_\mathrm{i} = N_\mathrm{m} N_\mathrm{o}$ and

$$\mathbf{x}_t = \left[\widetilde{\mathbf{S}}_{t, m-n_{\mathrm{m,L}}}, \ldots, \widetilde{\mathbf{S}}_{t, m+n_{\mathrm{m,R}}}\right]^T. \tag{15}$$

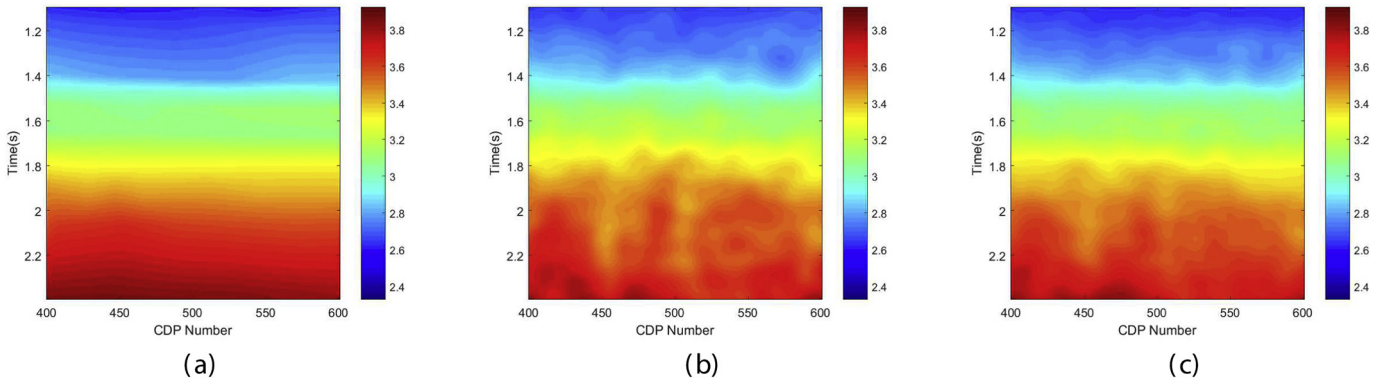

Fig. 4. Migration velocities - zoom into Fig. 2: (a) picked velocities used as ground truth; (b) predicted single channel velocities, $N_m = 1$, with 25% training; (c) predicted single channel velocities, $N_m = 1$ with 50% training.

During training, the output vector $\mathbf{z}_t$ is set to one expected velocity value ($M = 1$), such that $\mathbf{Z}$ is the corresponding velocity segment,

$$\mathbf{Z} = \left[\mathbf{V}_{k-(L_t-1),m}, \ldots, \mathbf{V}_{k,m}\right]^T. \tag{16}$$

Lastly, we ignore the first $L_t - 1$ values of the output $\mathbf{Z}$ and set the predicted velocity point value as the last one, i.e.,

$$\widetilde{\mathbf{V}}_{k,m} = \mathbf{z}_{L_t}. \tag{17}$$

The analysis volume moves through the seismic data and produces all expected velocity point values in the same manner. Each analysis volume, corresponding to a velocity segment, forms an instance of the net. The size and shape of the analysis volume defines the geometrical distribution of traces and samples to be used for each point's computation. The time window L_t is approximated considering the sampling rate and the sub-terrain characteristics to ensure that estimation of each point relies on suffice temporal information as well as spatial information.

The 2D output of the net $\widetilde{\mathbf{V}} \in \mathrm{R}^{L_s \times J_m}$ is filtered by a Gaussian 2D filter $\mathbf{H} \in \mathbb{R}^{N_G \times N_G}$ to yield the final predicted rms velocity field,

$$\widehat{\mathbf{V}}_{k,m} = \sum_{n,l} \mathbf{H}_{n,l} \widetilde{\mathbf{V}}_{k-n,m-l} \tag{18}$$

The solution could be generalized to 3D surveys with $\mathbf{V}_{3D} \in \mathbb{R}^{L_s \times J_{m_x} \times J_{m_y}}$ using a 4D analysis volumes of size $L_t \times N_{m_x} \times N_{m_y} \times N_o$. The analysis volume is then defined by N_{m_x} and N_{m_y}, the number of traces taken into account along the midpoint axes, N_o the number of offsets, and L_t time-depth samples along the vertical axis. It can be defined to associate with a point in its center, or in an asymmetrical manner. In a similar manner to the 2D configuration, for each velocity output point, the analysis 4D volume would be an instance input to the RNN. Moving the analysis volume along the 4D observation seismic data produces the entire 3D estimated rms velocity volume.

4. Real data results

In this section, we provide real data examples demonstrating the performance of the proposed technique. To implement the RNN we used TensorFlow (Abadi et al., 2015).

We apply the proposed method, to real seismic data from prestack 2D land vibroseis data provided by Geofizyka Torum Sp. Z.o.o Poland available in the public domain. Fig. 1 (a) and (b) show a horizontal

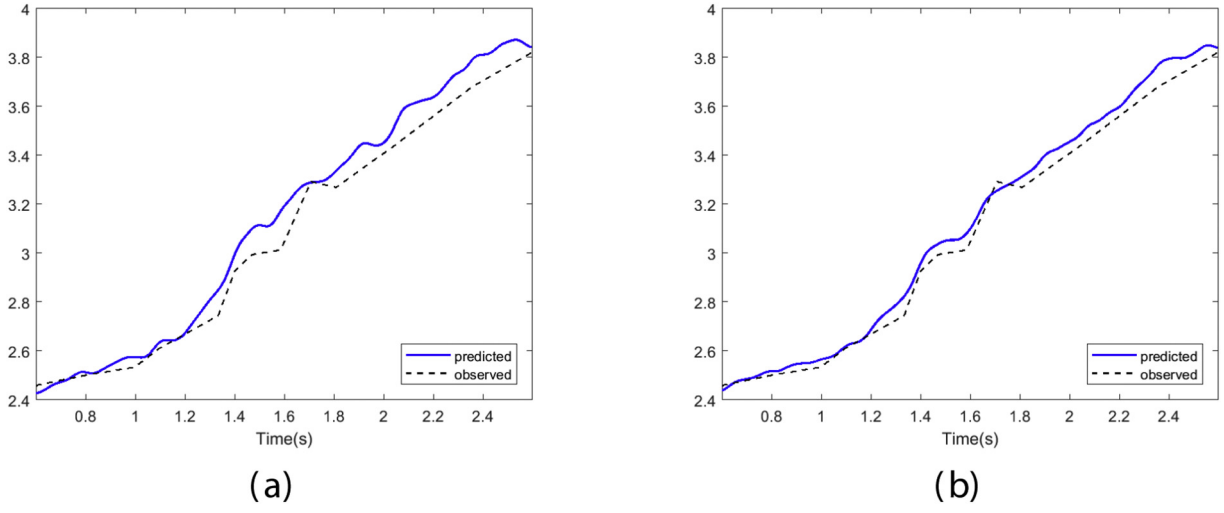

Fig. 5. Migration velocities for CMP index $m = 50$: (a) observed velocities vs. predicted single channel velocities, $N_m = 1$, with 25% training; (b) observed velocities vs. predicted single channel velocities, $N_m = 1$ with 50% training.

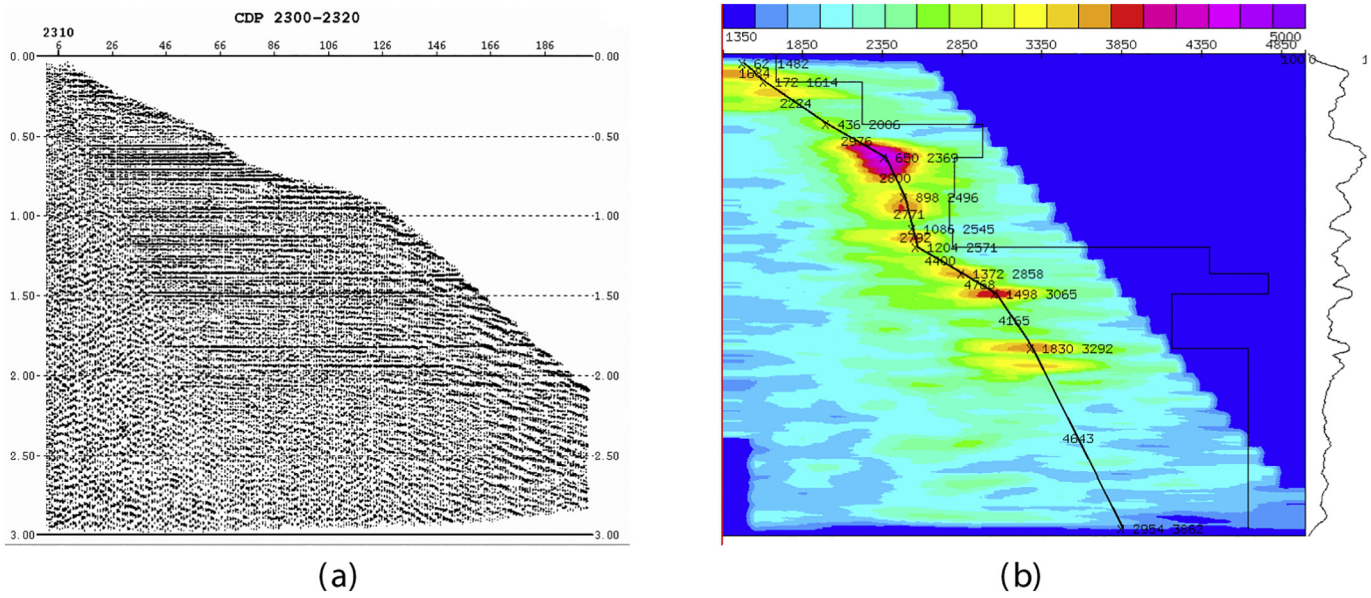

Fig. 6. Quality Control (QC) plots: (a) A super-gather for the CDP range 2300–2320 after application of NMO correction with the rms picked velocities. (b) Semblance values for rms migration velocities ranging from 1350 m/s to 5000 m/s. The picked velocities are marked with an x sign. The computed interval velocity is drawn with a solid black line to the right of the migration velocity picks.

slice and a vertical cross section through the seismic data, respectively. In order to avoid dead traces, we used 700 CDPs for demonstration. There are 71 offsets for each gather with offset total distance of 7 km. Each trace is 2.4 s in duration (600 samples). The time interval is 4 ms; offset spacing is 100 m; midpoint spacing is 125 m.

Migration velocity analysis was done by picking velocities from constant velocity migration gathers with 71 constant velocity panels, with velocities ranging from 1500 m/s to 5000 m/s. The picked migration velocities provided by GeoEnergy are depicted in Fig. 2(a). As can be seen, velocities in the region of interest are ranging between 2300 m/s and 4000 m/s. Since ground truth is unknown, for the sake of proof of concept, these velocities are treated as ground truth. The data is divided to a training set and a testing set such that 25% to 50% of the CMPs are allocated for training and the rest is used for testing. As a figure of merit we used the percentage error defined as

$$\mathbf{E}_{k,m} = \frac{|\mathbf{V}_{k,m} - \widehat{\mathbf{V}}_{k,m}|}{\mathbf{V}_{k,m}} \times 100\%.$$

Figs. 2(b), (c) show the results using one CMP for prediction, namely $N_m = 1$, with 25% and 50% training data, respectively. Window size of $L_t = 100$ time samples, and $N_o = 71$. Figs. 3(a), (b) plot the corresponding percentage error. As can be seen, the error is relatively low. Visually examining the estimated velocity and the percentage error, we observe that the predicted outcome generally fits the velocity field range and structure. On the other hand, the patterns lack some smoothness,

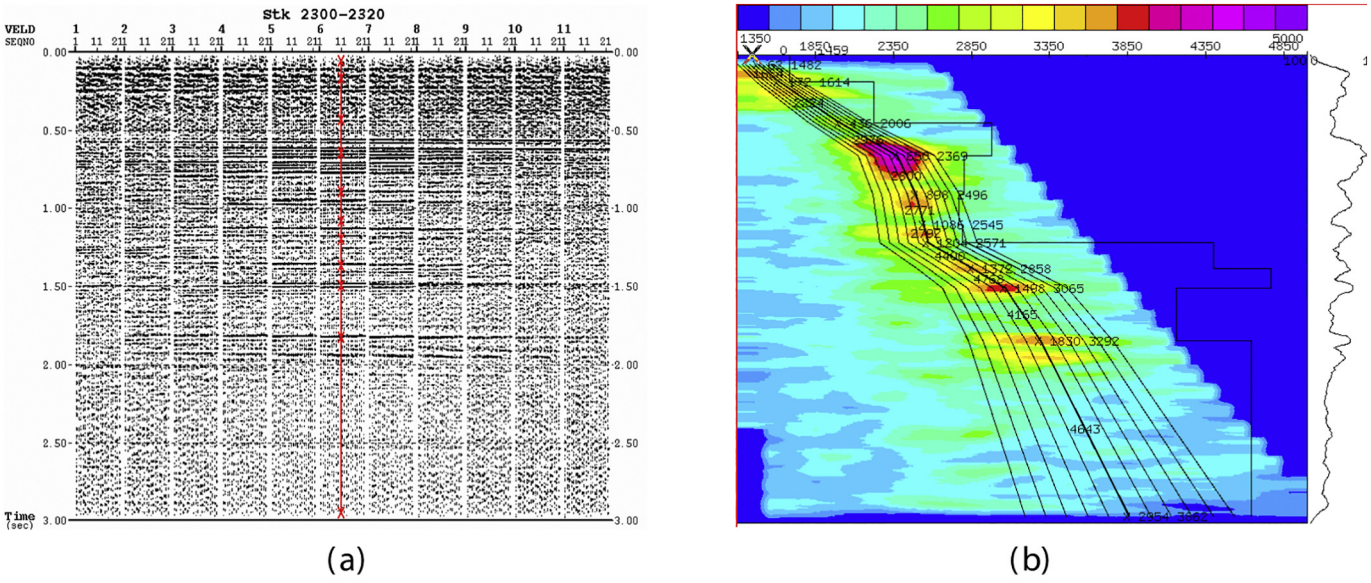

Fig. 7. Quality Control (QC) plots: (a) Migration stack of the supergather for the CDP range 2300–2320 for different velocities. (b) Semblance values for rms velocities. The migration velocities used for (a) are marked with a black solid line.

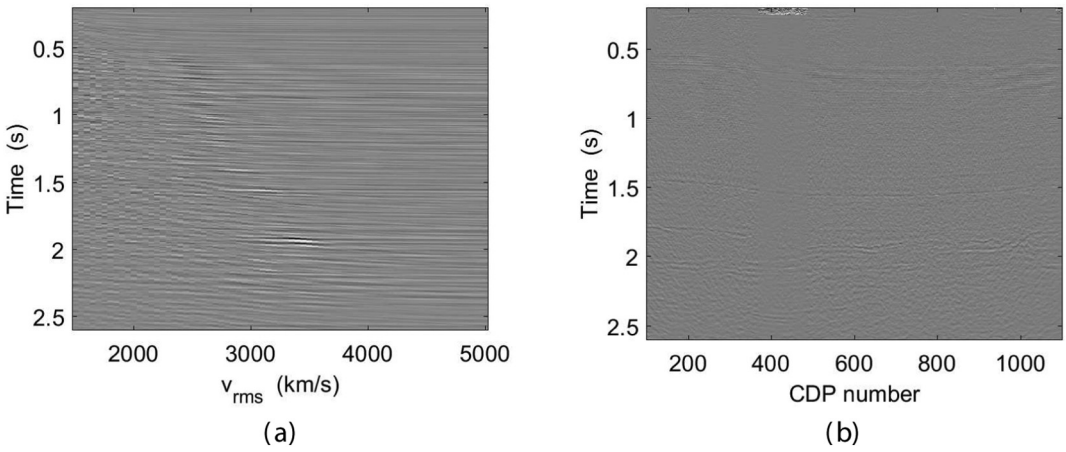

Fig. 8. Real seismic CVM panels: (a) a cross section at CDP index $m = 500$; (b) a cross section at a constant velocity of 4000 m/s.

since the relations between neighboring points along the midpoint direction are lost, as each CMP is processed separately. The error increases with depth, implying that the mapping between the seismic data to the velocity field may not be stationary as presumed. The experimental results affirm the intuitive assumption that as the percentage of data used for testing is larger, the percentage error is smaller. A zoom-into this example is depicted in Fig. 4. Figs. 5(a), (b) present the picked velocities, and the predicted velocities for CMP of index $m = 50$.

Fig. 2(d) shows the results using 3 CMPs for prediction, namely $N_m = 3$ and $n_{\mathrm{m,\,L}} = n_{\mathrm{m,\,R}} = 1$, with an analysis volume of size $100 \times 3 \times 71$. In other words, the velocity at each point is calculated taking into account the time window of the seismic data at the same CMP, and also at the preceding and at the consequent CMPs. The percentage error for this experiment is depicted in Fig. 3(c). As can be seen, in this example, increasing N_m, the number of CMPs in an analysis volume does not necessarily enhance the results. The size of the analysis cube, namely the number of offsets and CMPs, and the time window, is user dependent. Optimal results may vary depending on the data, sampling rate, offset spacing, midpoint spacing and more.

The proposed method's underlying assumption is that there exists a mapping $f: \mathbf{x} \rightarrow \mathbf{y}$ from the seismic acquired data to the migration velocity field, and that this mapping is the same for all analysis volumes across the seismic data to all velocity estimated points. This assumptions is, of course, controversial. As can be observed the percentage error deteriorates as we get into deeper layers.

To gain some insight into the challenges of MVA for the dataset we use in our experiments, we present a few of the quality-control (QC) preprocessing plots. Fig. 6(a) shows a super-gather for the CDP range 2300–2320 after application of NMO correction with the rms picked velocities. The starting CDP number is 1405 and the CDP number range is 1405–2685. As known, the quality of the picked stacking velocity is judged from the gather flatness after NMO correction (Öz Yilmaz et al., 2001). As can be seen, the gather reflection events are very flat after NMO correction. Fig. 6(b) shows the semblance values for rms migration velocities ranging from 1350 m/s to 5000 m/s. The picked velocities are marked with an x sign. The computed interval velocity is drawn with a solid black line to the right of the migration velocity picks.

Fig. 7(a) shows the migration stack of the supergather for the CDP range 2300–2320 for different migration velocities. The most left migrated stack, which has the least coherent signal energy, is the migrated stack resulting from the use of the most left migration velocity out the velocities marked with a black solid line in Fig. 7(b). The middle migration stack with the highest coherent signal energy corresponds to the picked velocity that is the middle velocity among the velocities. Note that the increasing envelope of the migration velocities with increasing time (depth) illustrates an increasing uncertainty in migration velocity picking with depth. Naturally, as we try to look deeper into the ground the reflection area gets smaller. In a sense, the measured sensitivity to changes recorded in the seismic data decreases with depth, which in term is expressed in less accurate velocity picking.

Typically, when training a neural net for common image processing applications, for example, training a classifier, usually about 20% of the data is used for testing, and the rest (70–80%) is used for training and validation. Then, the classifier can be applied to any data that is drawn from the same probability distribution. However, in the case of seismic MVA, obviously it would be impractical and inapplicable to use 70–80% of the dataset for training. Also, it is not possible to train a net with seismic data from one survey, and then apply the trained net to seismic data from a different survey, because we have no guarantee that data from a different survey is drawn from the same probability distribution. One can only apply the trained net to the data from the same survey that it was trained with. Hence, our objective is to train the net with a minimal percentage of data possible, and then apply the trained net to the rest of the data. So, in a sense, this is a few-shot learning problem (Fink, 2005). As shown in Sections 4 and 5, we check the performance of the model, by comparing the results, with different training percentage, to the picked velocities used as ground truths.

It should be noted that during DL training, specifically RNN training, there can be possible obstacles such as the well-known problems of vanishing gradients and exploding gradients (Glorot and Bengio, 2010; Géron, 2017). Generally speaking, DNNs tend to suffer from unstable gradients at the training stage. In our case study, in order to overcome these issues, it is recommended to choose L_t time steps that is not too large. Also, using batch normalization, and the use of a non-saturating activation function (such as the ReLU function) can mitigate these issues. Empirically, in our experiments, training was relatively stable.

5. Migration Velocity Picking using RNN

5.1. Seismic MVA using constant-velocity migration (CVM) panels

In order to pick migration velocities, either manually or by a neural net, one needs to work with constant velocity migrated gathers. The CVM volume associated with the data consists of J_{v} CVM panels, that is, prestack time migrated images generated by using a range of J_{v} constant velocities. Let us denote $\Phi \in \mathrm{R}^{L_{\mathrm{s}} \times J_{\mathrm{m}} \times J_{\mathrm{v}}}$ as a CVM volume associated with an observation seismic signal $\mathbf{S} \in \mathbb{R}^{L_{\mathrm{s}} \times J_{\mathrm{m}} \times J_{\mathrm{o}}}$, with J_{m} midpoints of L_{s} time samples, generated by J_{v} rms velocities. As before, we denote the ground truth prestack time migration velocity field as $\mathbf{V} \in \mathbb{R}^{L_{\mathrm{s}} \times J_{\mathrm{m}}}$, and the predicted velocity field as $\widehat{V} \in \mathrm{R}^{L_{\mathrm{s}} \times J_{\mathrm{m}}}$.

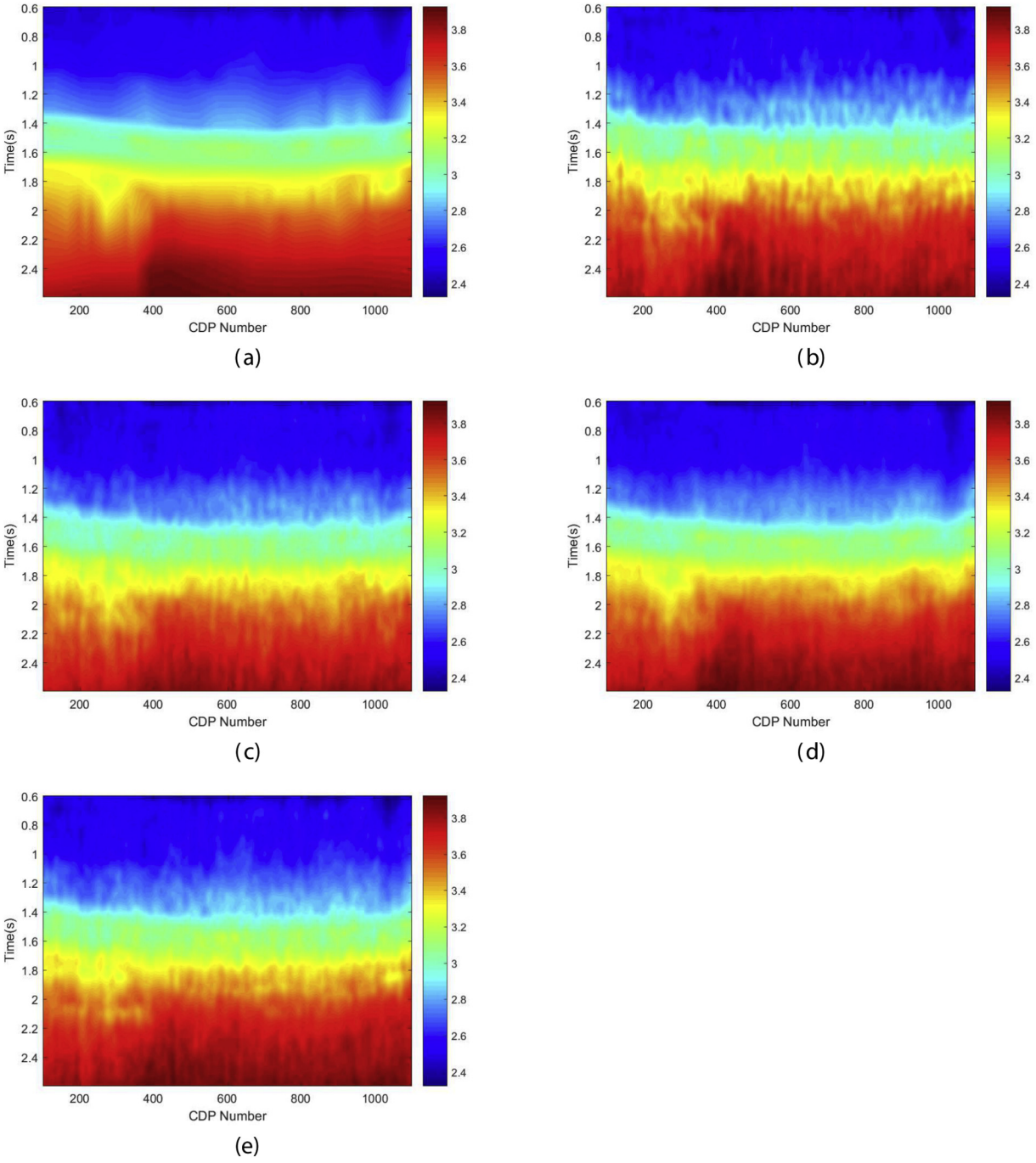

Fig. 9. MVA based on CVMs results: (a) picked velocities considered as ground truth; (b) predicted single channel velocities, $N_m = 1$, with 15% training; (c) predicted single channel velocities, $N_m = 1$ with 25% training; (d) predicted single channel velocities, $N_m = 1$, with 50% training; (e) predicted multichannel velocities, $N_m = 3$, with 25% training.

As previously stated, the rms velocity field is commonly built via picking of maximum amplitude horizon strands from the CVM panels, along the midpoint-time dimensions, keeping the corresponding velocity strands from the CVM volume. The horizon velocity strands are combined to construct the rms velocity field. In other words, the horizon picking during the constant migration velocity scanning is accomplished by locating the highest amplitudes related to a particular reflector that the geophysicist picking the migration velocities, sees in his or her workstation. This process is computationally demanding, takes a lot of time and requires manual work.

Now, by considering Φ as a preliminary given data, one can postulate that we merely replaced the given dataset $\mathbf{S}$ of coordinates midpoint-offset-two-way travel time with a dataset of axes midpoint-rms velocity-two-way zero-offset travel time (at event position after migration). In other words, we simply replaced the offset axis with the rms velocity axis. Therefore, we can use an RNN in the same manner as described in Section 3. Note that we expect improved performance in this case, since we reduced the burden of the learning task, by performing part of the velocity analysis process separately. Hereafter, we shall refer to the horizontal spatial position of the midpoint in the migrated domain as a constant depth point (CDP).

Definition 2 (Analysis CVM volume): We define an *analysis CVM volume* as a 3D volume of size $L_t \times N_{\mathrm{m}} \times N_{\mathrm{v}}$ enclosing L_{t} time (depth) samples of N_{m} CDPs, where each $L_t \times N_{\mathrm{m}} \times N_{\mathrm{v}}$ section consists of N_{v} patches of the migrated images corresponding to one of N_{v} constant migration velocities. Namely, we define a subvolume of the CVM volume, that

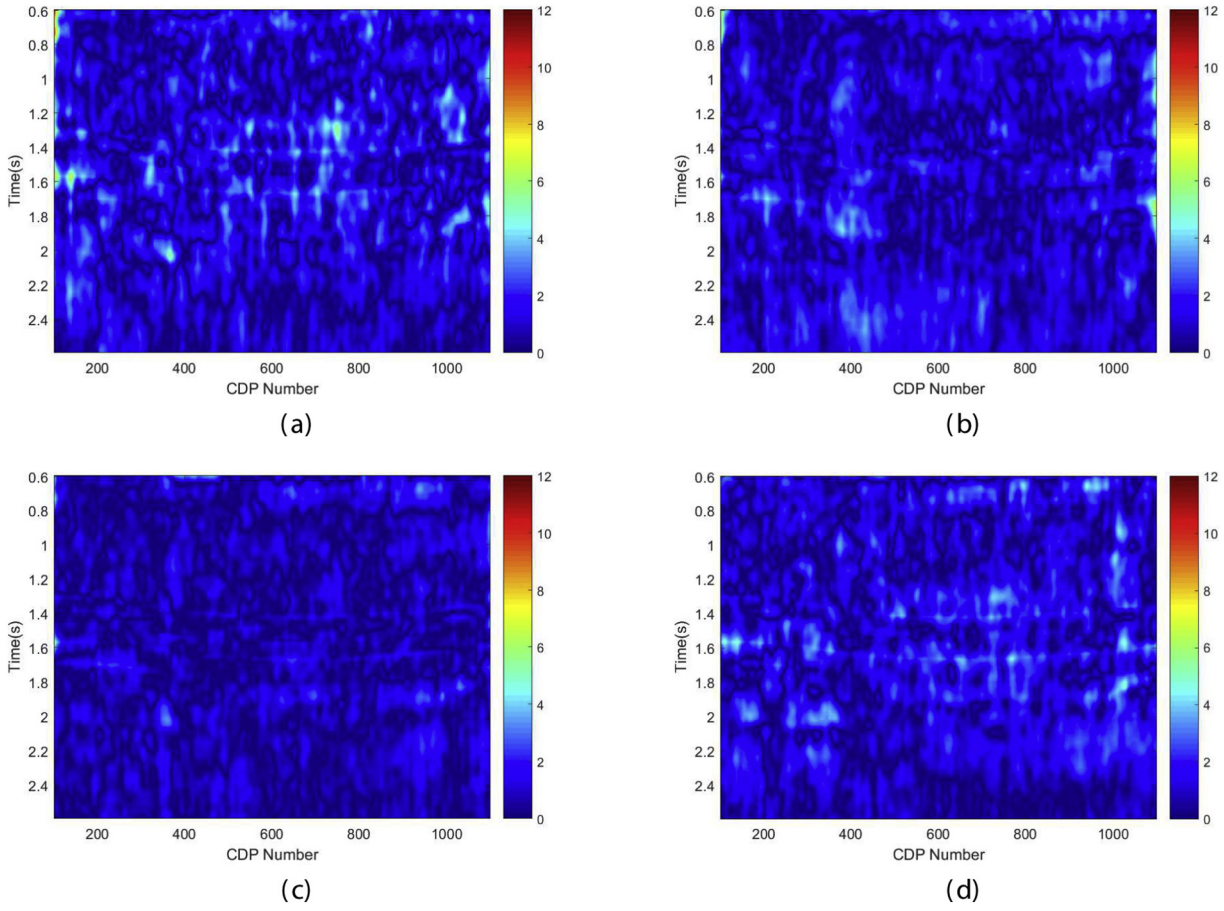

Fig. 10. Percentage error for MVA based on CVMs: (a) predicted single-channel velocities, $N_m = 1$ with 15% training; (b) predicted single-channel velocities, $N_m = 1$, with 25% training; (c) predicted single-channel velocities, $N_m = 1$, with 50% training; (d) predicted multichannel velocities, $N_m = 3$, with 25% training.

consists of a time window of L_t samples, N_m CDPs, generated by N_v velocities. We usually assume that $N_v = J_v$, such that we take into account the entire velocity range, but this can be adapted to a partial velocity range for different depths, which we will leave to further research. We assume $\{n_{m,L}, n_{m,R} \in \mathbb{N} : n_{m,L} + n_{m,R} = N_m - 1\}$ so that the analysis CVM volume associated with a point at time (depth) k at CDP index m is defined as

$$A_{k,m} = \begin{pmatrix} \widetilde{\Phi}_{k-L_t+1,m-n_{m,L}} & \cdots & \widetilde{\Phi}_{k-L_t+1,m+n_{m,R}} \\ \vdots & \ddots & \vdots \\ \widetilde{\Phi}_{k,m-n_{m,L}} & \cdots & \widetilde{\Phi}_{k,m+n_{m,R}} \end{pmatrix}, \tag{19}$$

where we denote $\widetilde{\Phi}_{k,m}$ as the corresponding midpoint-time data segment over all velocities, i.e.,

$$\widetilde{\Phi}_{k,m} = \left[\Phi_{k,m,1}, \ldots, \Phi_{k,m,J_v}\right]$$

As before, each analysis volume is normalized to have zero mean and unit variance.

An analysis CVM volume $A_{k,m}$ is related to a velocity field point $\mathbf{V}_{k,m}$. In a similar manner to (14), the input to the RNN is defined as

$$\mathbf{x} = A_{k,m}. \tag{20}$$

Hence, a time step input is essentially a data section of $n_i = N_m J_v$ CVM values at the corresponding time,

$$\mathbf{x}_t = \left[\widetilde{\Phi}_{tj-n_L}, \ldots, \widetilde{\Phi}_{tj+n_R}\right]^T. \tag{21}$$

Once again, we set the output vector $\mathbf{z}_t$ to one expected velocity value ($M = 1$), such that $\mathbf{Z}$ is the corresponding velocity segment,

$$\mathbf{Z} = \left[\mathbf{V}_{k-(L_t-1),m}, \ldots, \mathbf{V}_{k,m}\right]^T. \tag{22}$$

We ignore the first $L_t - 1$ values of the output $\mathbf{Z}$ and keep the last value as the predicted velocity point value,

$$\widetilde{V}_{k,m} = \mathbf{z}_{L_t}. \tag{23}$$

In order to produce all $L_s \times J_m$ predicted velocity point values, we slide the analysis CVM volume through the entire given CVM data. The time window L_t can be empirically determined while taking into account the sampling rate and the land characteristics. Usually setting N_m to 1 up to 3 CDPs in sufficient. In cases where the horizontal space is large it is recommended to limit the number of CDPs in the CVM analysis window to $N_m = 1$.

Finally, the filtered $\widetilde{V} \in \mathrm{R}^{L_s \times J_m}$ is computed as

$$\widehat{V}_{k,m} = \sum_{n,l} \mathbf{H}_{n,l} \widetilde{V}_{k-n,m-l}. \tag{24}$$

5.2. Application of RNN to real data CVM velocity picking

We apply the proposed RNN, to the real seismic data presented in Section 4. Here, we used 1000 CDPs in the migrated domain for demonstration. For each trace we processed 600 time samples (2.4 s in

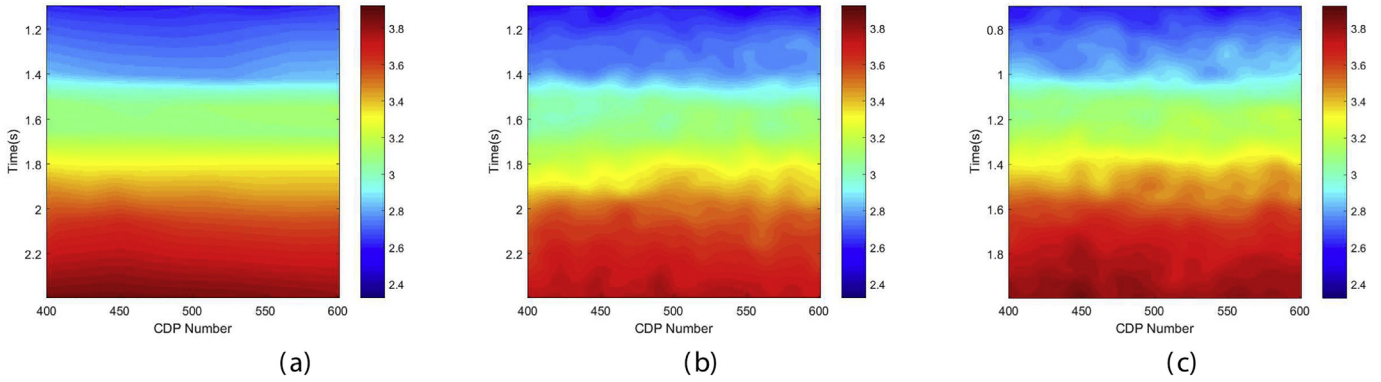

Fig. 11. Migration velocities - zoom into Fig. 9; (a) picked velocities used as ground truth; (b) predicted single channel velocities, $N_m = 1$, with 25% training; (c) predicted multichannel velocities, $N_m = 3$ with 25% training;

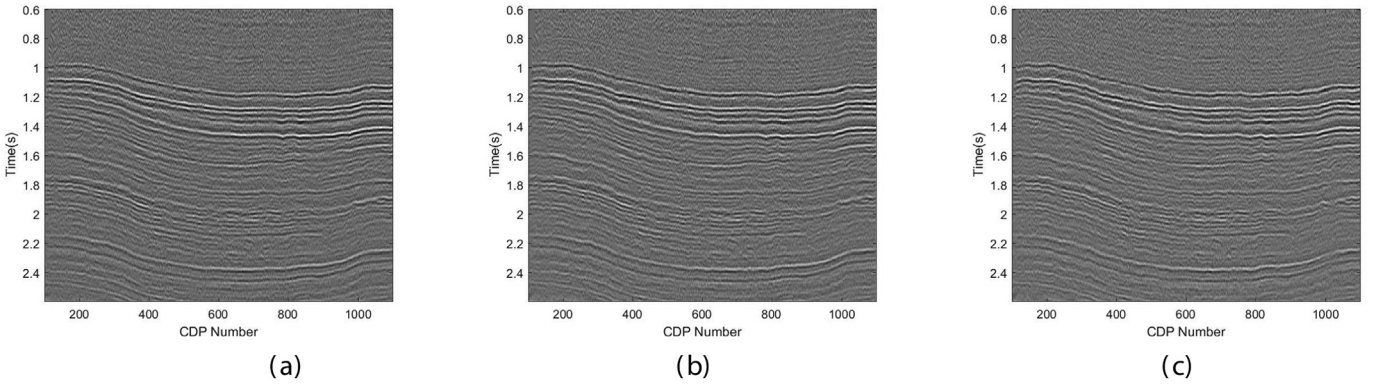

Fig. 12. PTSM based on CVMs MVA with: (a) picked velocities (courtesy of GeoEnergy); (b) predicted single channel velocities, $N_m = 1$, with 50% training; (c) predicted multichannel velocities, $N_m = 3$, with 25% training.

duration). The time interval is 4 ms; 71 offsets for each gather with offset total distance of 7 km. Offset spacing is 100 m; midpoint spacing is 125 m.

The CVM volume is composed of 71 vertical sections. Each of these sections corresponds to a constant migration velocity stack with the migration velocities starting at 1500 m/s and ending at 5000 m/s with a 50 m/s increment. Each of the 71 constant migration velocity stack sections has 1000 CDP points (in other words the spatial horizontal dimension is 1000) and 600 temporal samples with a 4 ms sampling rate. Fig. 8 shows the CVM volume associated with the seismic data. Fig. 8 (a) depicts a horizontal slice of the CVM volume, at CDP index $m = 500$, showing the migrated trace in time versus rms velocity. Fig. 8 (b) shows a vertical slice of the CVM volume at $v_{\mathrm{rms}} = 4000$ m/s, presenting the entire migrated image with one constant velocity.

As before, migration velocity analysis was done by picking velocities from constant velocity migration gathers with 71 constant velocity panels, with velocities ranging from 1500 m/s to 5000 m/s (Öz Yilmaz et al., 2001). The velocity field used as ground truth is depicted in Fig. 9(a). The data is divided to a training set and a testing set such that 15% to 50% of the CDPs are allocated for training and the rest is used for testing.

Note that the constant migration velocity increment is very small, only 50 m/s. If the neural net finds velocities between the 71 constant migration velocities (1500 to 5000 m/s with a 50 m/s velocity increment), it does not make any practical difference on the prestack time migrated stack.

The estimated rms velocities are presented in Fig. 9, for $N_m = 1, 3$ with different training percentage. Window size of $L_t = 100$ time samples, and $N_v = 71$. A zoom-in image is depicted in Fig. 11. The percentage error for these experiments can be shown in Fig. 10.

As can be observed, estimation based on CVM outperforms the estimation based solely on the seismic data (as presented in Section 4). In addition, the percentage of training data sufficient for errors smaller than 5% is significantly lower. As expected, increasing the percentage of data for training, and increasing N_m - the number of CDPs in the CVM analysis volume, improves the results, both in terms of percentage error and the structure and smoothness of the estimated velocity image.

Fig. 12 presents the prestack time migration (PTSM) image gathers generated by Kirchhoff migration, with the picked velocities compared to two examples of our estimated velocities, with $N_m = 1$ and 50% training, and $N_m = 3$ and 25% training. Visually examining the images, the differences between them can be hardly observed. As can be seen, the data is focused properly, and the shape and extent of the layers structure is preserved. It is safe to assume that the interpretation of the data will not be harmed. Hence, we have empirically verified that in spite of slight errors in the velocity estimation via the proposed RNN, comparing to the manually picked velocities, the degradation in the PTSM produced images is negligible.

Implementation of the above method is of relatively low computational complexity. Training on a standard workstation equipped with 32 GB of memory, an Intel(R) Core(TM) i7 - 9700K CPU @ 3.60 GHz, and an NVidia GeForce RTX 2080 Ti GPU, with 12 GB of video memory, converges after about 12,000 iterations in only 45 min. Whereas, manual velocity picking of the presented data required 8 h of work. Compared to current standard MVA methods, and to other DL training applications, that can take from days up to weeks, these are remarkable

results. Therefore, we postulate that the net is adequate for applications involving large volumes of data.

6. Conclusions

We presented two methods to perform MVA using RNN. To achieve this goal, we have assumed that there exists a mapping from each seismic data subvolume to a velocity point value in space. Hence, we have suggested to build an RNN that will attempt to learn this mapping, based on a *real data* training set. Then, the trained net computes the rest of the unknown velocity field. Alternatively, we have proposed to recast the RNN to learn a mapping from CVM panels to the velocity field.

The robustness of the proposed methods is validated via experimental real data results. We examined the role of the different parameters involved in the estimation, such as the training percentage of data, and the size of the analysis volume. As evident in the experimental results, a larger training set significantly reduces the percentage error. Whether there is a sufficient amount of training traces that suffices for a "good enough" velocity estimation is still an open question. We shall leave these issues and their theoretical analysis to future work. As expected, qualitative and quantitative assessment verify that velocity estimation from CVM panels is "easier" for the neural net. It is worth noting that the proposed schemes are designed to be as simple as possible, in order to make them adequate to handle immense amounts of real seismic data.

Future research can adapt the algorithm to non-stationary layers, incorporating different mappings from the data to different depth areas, or alternatively implement continual learning (Hsu et al., 2018) methods to the proposed scheme. In terms of the network architecture and training, to avoid overfitting, considering data augmentation, dropout (Srivastava et al., 2014) and skipping connections (Orhan and Pitkow, 2017) can also be examined.

Declaration of Competing Interest

None.

Acknowledgment

The authors thank the associate editor and the anonymous reviewers for their constructive comments and useful suggestions.

Declaration of Competing Interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Man, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Vigas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X., 2015. Tensorflow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. URL. http://download.tensorflow.org/paper/whitepaper2015.pdf.

Araya-Polo, M., Dahlke, T., Frogner, C., Zhang, C., Poggio, T., Hohl, D., 2017. Automated fault detection without seismic processing. Lead. Edge 36, 208–214.

Berkhout, A., 1986. The seismic method in the search for oil and gas: current techniques and future development. Proc. IEEE 74, 1133–1159.

Beylkin, G., 1985. Imaging of discontinuities in the inverse scattering problem by inversion of a causal generalized radon transform. J. Math. Phys. 26, 99–108.

Biondi, B., 2006. 3D Seismic Imaging. Society of Exploration Geophysicists.

Biswas, R., Vassiliou, A., Stromberg, R., Sen, M.K., 2018. Stacking velocity estimation using recurrent neural network. SEG Technical Program Expanded Abstracts. 2018, pp. 2241–2245.

Bleinstein, N., 1984. Mathematical Methods for Wave Phenomena. Academic Press.

Calderón-Maciás, C., Sen, M.K., Stoffa, P.L., 1998. Automatic NMO correction and velocity estimation by a feedforward neural network. Geophysics 63, 1696–1707.

Chen, G., 2016. A Gentle Tutorial of Recurrent Neural Network with Error Backpropagation. ArXiv e-prints arXiv 1610.02583.

Cohen, J.K., Bleistein, N., 1979. Velocity inversion procedure for acoustic waves. Geophysics 44, 1077–1087.

Dix, C.H., 1955. Seismic velocities from surface measurements. Geophysics 20, 68–86.

Dorrington, K.P., Link, C.A., 2004. Genetic algorithm\neural-network approach to seismic attribute selection for welllog prediction. Geophysics 69, 212–221.

Fink, M., 2005. Object classification from a single example utilizing class relevance metrics. Advances in Neural Information Processing Systems 17. MIT Press, pp. 449–456.

Fish, B.C., Kusuma, T., 1994. A neural network approach to automate velocity picking. SEG Technical Program Expanded Abstracts, pp. 185–188.

Géron, A., 2017. Hands-on Machine Learning with Scikit-Learn and TensorFlow : Concepts, Tools, and Techniques to Build Intelligent Systems. O'Reilly Media.

Glorot, X., Bengio, Y., 2010. Understanding the difficulty of training deep feedforward neural networks. J. Mach. Learn. Res. Proc. Track 9, 249–256.

Hagedoorn, J.G., 1954. A process of seismic reflection interpretation. Geophys. Prospect. 2, 85–127.

Haris, A., Murdianto, B., Susattyo, R., Riyanto, A., 2018. Transforming seismic data into lateral sonic properties using artificial neural network: a case study of real data set. Int. J. Technol. 9, 472.

Haris, A., Sitorus, R., Riyanto, A., 2017. Pore pressure prediction using probabilistic neural network: case study of South Sumatra basin. IOP Conf. Series Earth Environ. Sci. 62, 012021.

Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Comput. 9, 1735–1780.

Hollander, Y., Merouane, A., Yilmaz, O., 2018. Using a deep convolutional neural network to enhance the accuracy of first-break picking. SEG Technical Program Expanded Abstracts, pp. 4628–4632.

Hsu, Y., Liu, Y., Kira, Z., 2018. Re-evaluating continual learning scenarios: A categorization and case for strong baselines. NeurIPS Continual Learning Workshop.

Kahrizi, A., Hashemi, H., 2014. Neuron curve as a tool for performance evaluation of MLP and RBF architecture in first break picking of seismic data. J. Appl. Geophys. 108, 159–166.

Kumar, P.C., Omosanya, K.O., Alves, T.M., Sain, K., 2019a. A neural network approach for elucidating fluid leakage along hard-linked normal faults. Mar. Pet. Geol. 110, 518–538.

Kumar, P.C., Omosanya, K.O., Sain, K., 2019b. Sill cube: an automated approach for the interpretation of magmatic sill complexes on seismic reflection data. Mar. Pet. Geol. 100, 60–84.

Kumar, P.C., Sain, K., 2018. Attribute amalgamation-aiding interpretation of faults from seismic data: an example from waitara 3d prospect in Taranaki basin off New Zealand. J. Appl. Geophys. 159, 52–68.

Kumar, P.C., Sain, K., Mandal, A., 2019c. Delineation of a buried volcanic system in kora prospect off New Zealand using artificial neural networks and its implications. J. Appl. Geophys. 161, 56–75.

Li, S., Liu, B., Ren, Y., Chen, Y., Yang, S., Wang, Y., Jiang, P., 2019. Deep learning inversion of seismic data. CoRR abs/1901.07733. URL. http://arxiv.org/abs/1901.07733 arXiv:1901.07733.

McCormack, M.D., Rock, A.D., 1993. Adaptive Network for Automated First Break Picking of Seismic Refraction Events and Method of Operating the Same.

Mężyk, M., Malinowski, M., 2019. Multi-pattern algorithm for first-break picking employing open-source machine learning libraries. J. Appl. Geophys. 170, 103848.

Murat, M.E., Rudman, A.J., 1992. Automated first arrival picking: a neural network approach. Geophys. Prospect. 40, 587–604.

Orhan, A.E., Pitkow, X., 2017. Skip connections eliminate singularities. ArXiv e-prints arXiv:1701.09175.

Pereg, D., Cohen, I., Vassiliou, A., 2020. Sparse seismic deconvolution via recurrent neural network. J. Appl. Geophys. 175, 103979.

Sava, P., Biondi, B., 2004. Wave-equation migration velocity analysis. I. Theory. Geophys. Prospect. 52, 593–606.

Schleicher, J., Tygel, M., Hubral, P., 1993. 3D true-amplitude finite offset migration. Geophysics 1112–1126.

Schneider, W.A., 1978. Integral formulation for migration in two and three dimensions. Geophysics 43, 49–76.

Sheriff, R., Geldart, L., 1983. Exploration Seismology. Cambridge University Press, UK.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. 15, 1929–1958.

Wang, W., Yang, F., Ma, J., 2018. Automatic salt detection with machine learning, in: 80th European Association of Geoscientists and Engineers Conference and Exhibition Extended Abstracts. p. 912.

Yilmaz, Öz, Tanir, I., Gregory, C., 2001. A unified 3D seismic workflow. Geophysics 66, 1699–1713.

Zhang, C., Frogner, C., Araya-Polo, M., Hohl, D., 2014. Machine-learning based automated fault detection in seismic traces. Proceedings of 76th EAGE Conference and Exhibition.