Jump-starting neural network training for seismic problems

Fantine Huot

ABSTRACT

Deep learning algorithms are immensely data-hungry and rely on large amounts of labeled data to achieve good performance. However the earth is intrinsically unlabeled and we are often confronted to fuzzy boundaries, uncertain labels, and absence of ground truth. Moreover, deep learning models do not always generalize well to conditions that are different from the ones encountered during training. In this context, it can be difficult to leverage deep learning algorithms for seismic problems. Herein we introduce strategies for overcoming these limitations, using synthetic data generation and transfer learning to jump-start the training of neural networks. We present this methodology through two case studies: earthquake detection using the Northern California Seismic Network (NCSN); and targeted noise filtering for ambient seismic noise recorded by a fiber optic array underneath Stanford campus.

INTRODUCTION

Deep neural networks can achieve accurate mappings from inputs to outputs from large amounts of labeled data. Within the last 5 years, deep learning has had a dramatic impact on computer vision (Krizhevsky et al., 2012; He et al., 2015), speech recognition (Dahl et al., 2012; Deng et al., 2010; Seide et al., 2011; Hinton et al., 2012), and image segmentation (Sermanet et al., 2013; Farabet et al., 2013; Couprie et al., 2013; Cireşan et al., 2012).

However, these models are immensely data-hungry and rely on huge amounts of labeled data to achieve their performance. As of 2016, a rough rule of thumb is that a supervised deep learning algorithm will generally achieve acceptable performance with around 5,000 labeled examples per category and will match or exceed human performance when trained with a dataset containing at least 10 million labeled examples (Goodfellow et al., 2016).

One of the challenges with applying deep learning algorithms to seismic problems is that we are often confronted to limited labeled data. The earth is intrinsically unlabeled, and we have to deal with uncertain labels, fuzzy boundaries, and absence of ground truth. Seismic datasets also tend to be unbalanced, as we are trying to detect rare and sparse events within large amounts of background noise.

Another problem is that deep learning models do not always generalize well to conditions that are different from those encountered during training on a carefully constructed dataset. Seismic field datasets have low signal to noise ratio, and suffer from missing data and dropouts. They present coherent noise sources and artifacts, many of which the deep learning model has not encountered during training.

In this study we present how synthetic data generation and transfer learning can jump-start the training of neural networks in order to overcome these limitations. We outline reasons why transfer learning warrants our attention and describe different transfer learning scenarios. We then illustrate this methodology through two case studies: earthquake detection using the Northern California Seismic Network (NCSN); and targeted noise filtering for ambient seismic noise recorded by a fiber optic array underneath Stanford campus.

SYNTHETIC DATA GENERATION

When manually labeling the data is infeasible, how can we come up with large volumes of labeled data? The key idea here is to leverage our domain-knowledge to generate synthetic data to boost our training data.

The use of synthetically generated training data is pretty common in machine learning and has been successfully used for various applications, ranging from character recognition in natural images (de Campos et al., 2009), traffic sign recognition (Greenhalgh and Mirmehdi, 2012; Hoessler et al., 2007) or handwriting recognition (Varga and Bunke, 2003), to more elaborate problems such as face recognition (Fanelli et al., 2011) or protein interactions (Pham and Jain, 2006). Synthetic data generation is also used for unbalanced classification problems to ensure that each class is sufficiently well represented to obtain a classification system of high generalization performance (Fanelli et al., 2011; Zadrozny, 2004). The synthetic training examples do not per se have to capture all the complexities of the field data. The DeepSketch2Image project (Seddati et al., 2016) demonstrates it is possible to classify photos even by training on free-hand doodles.

For physical modeling problems, neural networks are often trained solely on synthetic data, and have had successful results for various applications such as turbulent flow modeling Ling et al. (2016), or modeling error estimation Trehan et al. (2017).

TRANSFER LEARNING

The ability to transfer knowledge to new conditions is generally known as transfer learning.

In the classic supervised learning scenario, we train a neural network for a task and domain A for which we have large amounts of labeled data, and expect it to perform well on unseen data of the same task and domain. Given some other related task or domain B, the classic supervised learning paradigm breaks down when we do not have sufficient labeled data to train a reliable model.

Transfer learning allows us to deal with this scenario by leveraging the already existing labeled data of the first task or domain A, storing the knowledge gained in solving this task, and applying it to our related problem of interest B. This knowledge transfer can be achieved through various transfer learning scenarios (Pan and Yang, 2010; Yosinski et al., 2014):

- Using the pre-trained network as a fixed feature extractor (Razavian et al., 2014): We pre-train the convolutional neural network (CNN) on dataset A, remove the last fully-connected layer, and treat the rest of the network as a fixed feature extractor for the new dataset B.
- Fine-tuning the pre-trained network: We pre-train the CNN on dataset A, and fine-tune its weights using dataset B. It is possible to fine-tune all the layers of the network, or to keep some of the earlier layers fixed (due to overfitting concerns) and only fine-tune some higher-level portion of the network. This is motivated by the observation that the earlier features of a CNN contain more generic features (e.g. edge detectors or color blob detectors) that should be useful to many tasks, but later layers of the network becomes progressively more specific to the details of the classes contained in the original dataset.
- Using pre-trained models: Since large modern convolutional networks take 2-3 weeks to train across multiple GPUs, pre-trained network weights are often released publicly for others to use.

A description of the different components of a convolutional network is provided in Huot (2018). Which type of transfer learning to use is a function of several factors, but typically depends on the size of the dataset for task B and its similarity to the dataset A. In the following sections, we present some applications of this methodology.

EARTHQUAKE DETECTION USING THE NORTHERN CALIFORNIA SEISMIC NETWORK (NCSN)

We used the Northern California Seismic Network (NCSN) catalog of earthquake detections to build a dataset of labeled earthquake and background noise waveforms. In this study we used data from the following seismic stations (Figure 1):

- BK-SAO, San Andreas Geophysical Observatory, Hollister,
- BK-JRSC, Jasper Ridge Biological Preserve, near Stanford,
- BK-PKD, Bear Valley Ranch, Parkfield,

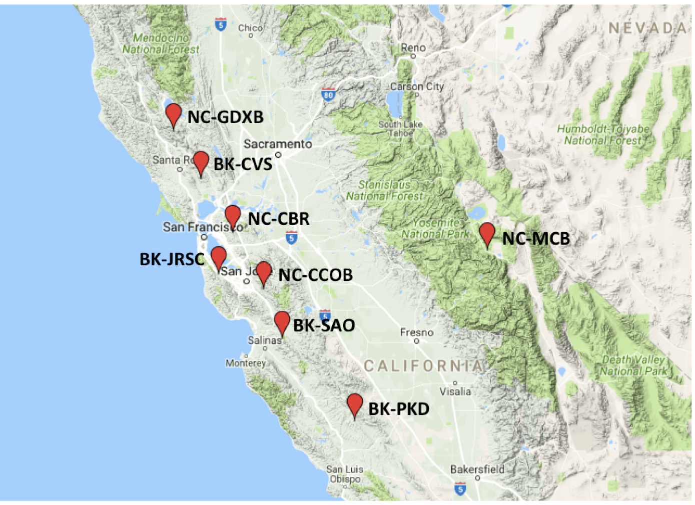
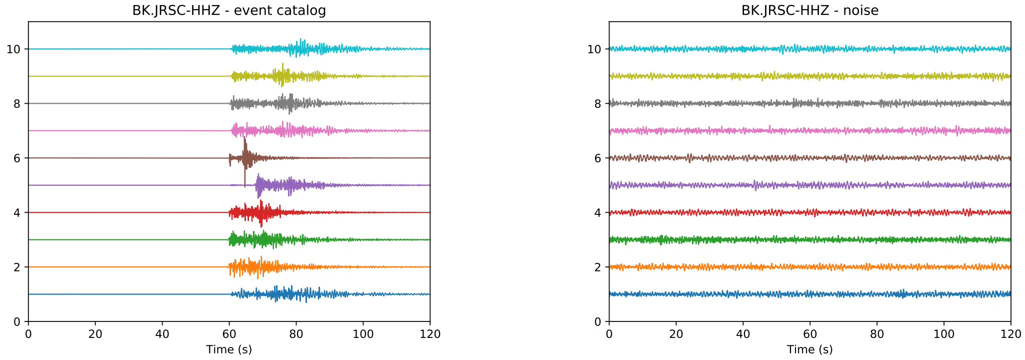


Figure 1: Geographical locations of selected seismic stations from the NCSN. [**NR**]

- BK-CVS, Carmenet Vineyards, Sonoma.

These stations are equipped with 3-component broadband seismometers. For each station, we extracted the earthquake waveforms from the catalog from 01/01/2008 to 06/01/2016 for events of magnitude > 2.0. The background noise waveforms were picked at random times for each day in the time period, while ensuring that they did not overlap with any event from the catalog and did not trigger the STA/LTA (Short Term Average/Long Term Average) event detection algorithm (with STA window of 3s, LTA window of 45s, and threshold of 6.0).

All the waveforms were bandpassed between 1 and 10 Hz, downsampled to a 20 Hz sampling rate, and then normalized (Figure 2).



(a)

(b)

Figure 2: Example waveforms of (a) earthquakes and (b) background noise recorded at the BK-JRSC station. [**ER**]

To capture the temporal variations of each waveform, we decomposed the signal into wavelet attributes by applying continuous wavelet transforms (CWT). CWT are frequently used in pattern recognition to decompose complex patterns into elementary forms by comparing the input signal to shifted and compressed or stretched versions of an analyzing wavelet (Mallat, 2008).

We used the Morlet wavelet as the analyzing function and subsequently took the amplitude of the resulting complex numbers. For each waveform we computed 30 CWT scale factors. At this stage, we subsampled the data by averaging the CWT scales over windows of 0.5s, and the time windows were then narrowed down to a 15s detection window. This resulted in 30 × 30 attributes for each waveform (Figures 3a and 3b). Each attribute was standardized to zero mean and unit variance. Means and standard deviations were stored for later use when applying the trained algorithm to new data.

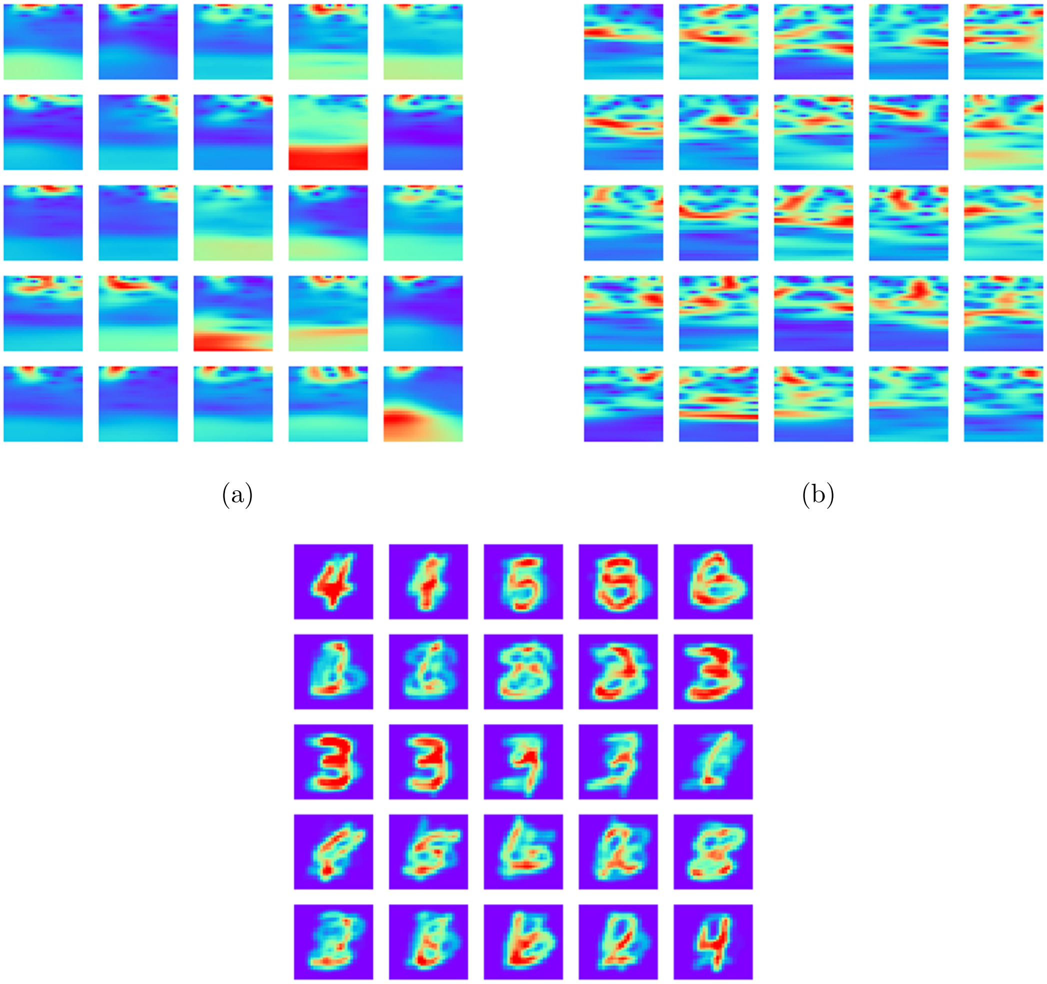
After pre-processing, we ended up with about 2,000 earthquake samples and 3,000 background noise samples for each station. The dataset was then separated according to a 80:20 ratio into a training set and a test set.

We trained a small CNN with 2 convolutional layers and a fully connected output layer with a softmax classifier on this dataset. However, as the dataset was fairly small, the network achieved poor accuracy. We then performed transfer learning using the the MNIST dataset. The MNIST database (Modified National Institute of Standards and Technology database) is a large database of handwritten digits that is commonly used for training various image processing systems (Fig 3c). It is widely used for training and testing in the field of machine learning. While the classification of handwritten digits has very little to do with earthquake detection, the basic image processing performed in the convolutional layers has similar behavior, and acts as a feature extractor. By pre-training using the MNIST dataset, we can use transfer learning by copying the weights of the convolutional layers, while leaving out the classification part. This process allows us to initialize the weights in our earthquake detection network to a reasonable initial guess.

Using this approach, the network obtained 99.5% accuracy when trained, and tested on only one of the stations. When mixing all four stations, its accuracy dropped to 96.8%. Although this result is not as good, it shows that it might be possible to generalize to more stations.

TRAFFIC DETECTION FOR TARGETED FILTERING FOR FIBER OPTIC AMBIENT SEISMIC NOISE

Distributed acoustic sensing (DAS) is an emerging technology used to record seismic data that employs fiber optic cables as a probing system. Recently, a DAS array has been deployed beneath Stanford campus in the existing fiber optic telecommunication conduits. As we can so easily use our telecomm infrastructure for continuous, dense, seismic acquisition, data collected in such a manner will go to waste unless we



(a)

(b)

(c)

Figure 3: Example waveform attributes computed for (a) earthquakes and (b) background noise. (c) Examples from the MNIST hand-written numbers dataset. [**CR**]

significantly automate ambient noise processing. In particular, coherent noise sources inhibit reliable extraction of useful signals (Martin et al., 2016). Herein, we train a convolutional neural network for detecting traffic noise in order to selectively filter it out to generate ambient seismic noise fields that are suitable for interferometry purposes.

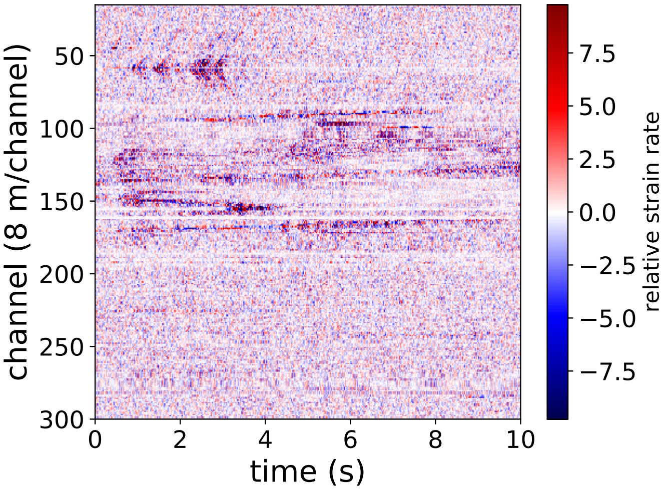


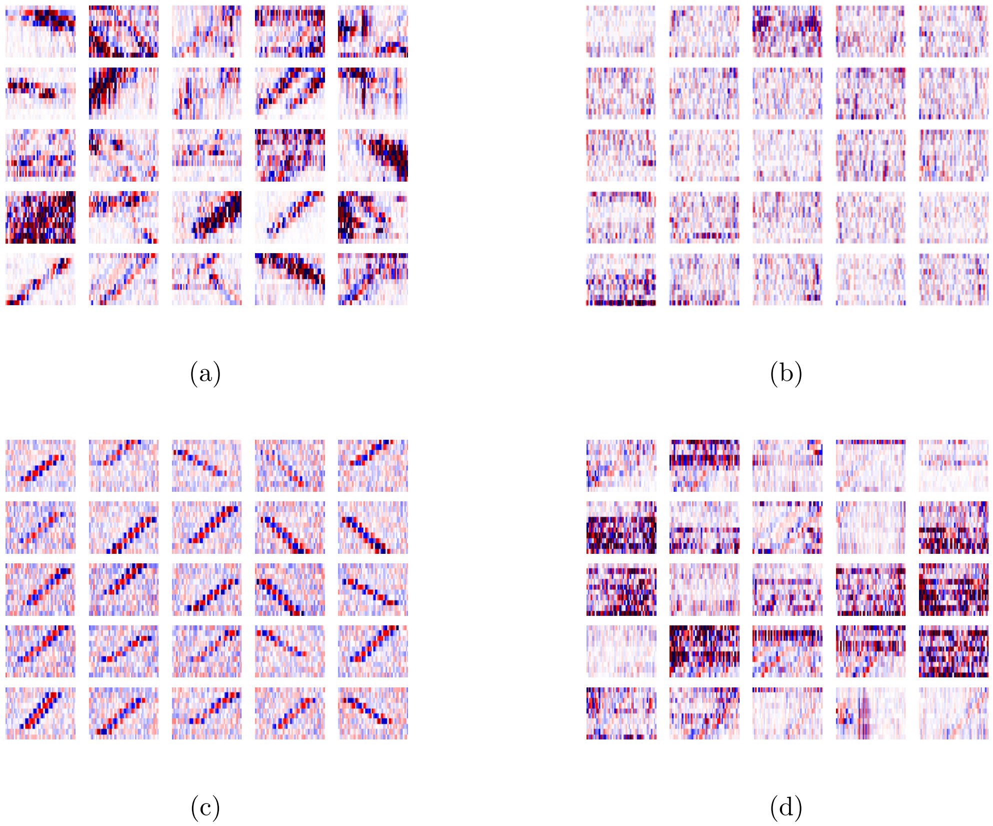
Figure 4: Data recorded by the Stanford DAS array. Ambient noise in urban areas is far from white or spatially uniform. [**ER**]

Every 8m of fiber acts as a seismic sensor, and the DAS array generates contiguous time series that conveniently lend themselves to image processing (Figure 4). We used a previously trained unsupervised learning approach to identify traffic and background noise (Huot et al., 2017; Martin et al., 2018). We opted for a detection window of 10 channels by 10 seconds. The windows were downsampled along the time axis, to 10 × 50.

The dataset was fairly small and highly imbalanced, as there were far fewer car events than background noise. Therefore, we boosted the dataset by generating simplistic synthetic car data (Figure 5c). All the windows were shuffled to avoid any time bias and normalized with the same means and standard deviations. The data sets were then arranged as follows:

- The training data contained 50,000 windows, 50% noise, 50% cars. 80% of these cars were synthetically generated.
- The testing data contained 10,000 windows, 50% noise, 50% cars, but only real data.

We then designed a small CNN with 2 convolution layers with long stencils on the time axis.



(a)

(b)

(c)

(d)

Figure 5: 10 channels × 10 seconds data windows with (a) traffic noise. (b) background data. (c) simplistic synthetically generated traffic noise. (d) examples of traffic noise to which the network gave lower classification probabilities. [**CR**]

This classifier achieved 99.4% accuracy on this dataset. Out of 5000 cars, 38 obtained a probability score less than 90%, 59 less than 95%. Despite having been trained using mostly synthetic car data, the network performed well at detecting traffic noise, even on examples with faint car events or very noisy data portions (Figure 5d).

DISCUSSION AND CONCLUSIONS

Through two case studies, earthquake detection and traffic noise detection, we introduced strategies to jump-start the training of neural networks when confronted to limited amount of labeled data and unbalanced datasets. In particular, we demonstrated that transfer learning and synthetic data generation allow us to leverage domain knowledge and pre-train the network using simulated data. Going forward, we aim to apply this methodology to more use cases such as stratigraphy estimation and seismic imaging.

ACKNOWLEDGEMENTS

We thank Karianne Bergen and Eileen R. Martin for their advice for processing the NCSN earthquake catalog and the Stanford DAS array data, respectively.

REFERENCES

Cireşan, D., U. Meier, J. Masci, and J. Schmidhuber, 2012, Multi-column deep neural network for traffic sign classification: Neural Networks, **32**, 333–338.

Couprie, C., C. Farabet, L. Najman, and Y. LeCun, 2013, Indoor semantic segmentation using depth information: arXiv preprint arXiv:1301.3572.

Dahl, G. E., D. Yu, L. Deng, and A. Acero, 2012, Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition: IEEE Transactions on Audio, Speech, and Language Processing, **20**, 30–42.

de Campos, T. E., B. R. Babu, and M. Varma, 2009, Character recognition in natural images.: VISAPP (2), 273–280.

Deng, L., M. L. Seltzer, D. Yu, A. Acero, A.-R. Mohamed, and G. E. Hinton, 2010, Binary coding of speech spectrograms using a deep auto-encoder.: Interspeech, Citeseer, 1692–1695.

Fanelli, G., J. Gall, and L. Van Gool, 2011, Real time head pose estimation with random regression forests: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, IEEE, 617–624.

Farabet, C., C. Couprie, L. Najman, and Y. LeCun, 2013, Learning hierarchical features for scene labeling: IEEE transactions on pattern analysis and machine intelligence, **35**, 1915–1929.

Goodfellow, I., Y. Bengio, and A. Courville, 2016, Deep learning: MIT Press.

Greenhalgh, J., and M. Mirmehdi, 2012, Real-time detection and recognition of road traffic signs: IEEE Transactions on Intelligent Transportation Systems, **13**, 1498–1506.

He, K., X. Zhang, S. Ren, and J. Sun, 2015, Delving deep into rectifiers: Surpassing human-level performance on imagenet classification: Proceedings of the IEEE international conference on computer vision, 1026–1034.

Hinton, G., L. Deng, D. Yu, G. E. Dahl, A.-R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, et al., 2012, Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups: IEEE Signal Processing Magazine, **29**, 82–97.

Hoessler, H., C. Wöhler, F. Lindner, and U. Kreßel, 2007, Classifier training based on synthetically generated samples: Presented at the Proceedings of 5th international conference on computer vision systems. Bielefeld, Germany.

Huot, F., 2018, Convolutional neural networks explained: SEP-Report, **172**.

Huot, F., Y. Ma, R. Cieplicki, E. R. Martin, and B. Biondi, 2017, Automatic noise exploration in urban areas, *in* SEG Technical Program Expanded Abstracts 2017: Society of Exploration Geophysicists, 5027–5032.

Krizhevsky, A., I. Sutskever, and G. E. Hinton, 2012, Imagenet classification with deep convolutional neural networks: Advances in neural information processing systems, 1097–1105.

Ling, J., A. Kurzawski, and J. Templeton, 2016, Reynolds averaged turbulence modelling using deep neural networks with embedded invariance: Journal of Fluid Mechanics, **807**, 155–166.

Mallat, S., 2008, A wavelet tour of signal processing: the sparse way: Academic press.

Martin, E. R., F. Huot, Y. Ma, R. Cieplicki, S. Cole, M. Karrenbach, and B. L. Biondi, 2018, A seismic shift in scalable acquisition demands new processing: Fiber-optic seismic signal retrieval in urban areas with unsupervised learning for coherent noise removal: IEEE Signal Processing Magazine, **35**, 31–40.

Martin, E. R., N. Lindsey, S. Dou, J. Ajo-Franklin, T. Daley, B. Freifeld, M. Robertson, C. Ulrich, A. Wagner, K. Bjella, et al., 2016, Interferometry of a roadside das array in fairbanks, ak: Presented at the 2016 SEG International Exposition and Annual Meeting, Society of Exploration Geophysicists.

Pan, S. J., and Q. Yang, 2010, A survey on transfer learning: IEEE Transactions on knowledge and data engineering, **22**, 1345–1359.

Pham, T. A., and A. N. Jain, 2006, Parameter estimation for scoring protein- ligand interactions using negative training data: Journal of medicinal chemistry, **49**, 5856–5868.

Razavian, A. S., H. Azizpour, J. Sullivan, and S. Carlsson, 2014, Cnn features off-the-shelf: an astounding baseline for recognition: Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on, IEEE, 512–519.

Seddati, O., S. Dupont, and S. Mahmoudi, 2016, Deepsketch2image: Deep convolutional neural networks for partial sketch recognition and image retrieval: Proceedings of the 2016 ACM on Multimedia Conference, ACM, 739–741.

Seide, F., G. Li, and D. Yu, 2011, Conversational speech transcription using context-dependent deep neural networks.: Interspeech, 437–440.

Sermanet, P., K. Kavukcuoglu, S. Chintala, and Y. LeCun, 2013, Pedestrian detection with unsupervised multi-stage feature learning: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 3626–3633.

Trehan, S., K. Carlberg, and L. J. Durlofsky, 2017, Error modeling for surrogates of dynamical systems using machine learning: International Journal for Numerical Methods in Engineering.

Varga, T., and H. Bunke, 2003, Generation of synthetic training data for an hmm-based handwriting recognition system: Document Analysis and Recognition, 2003. Proceedings. Seventh International Conference on, IEEE, 618–622.

Yosinski, J., J. Clune, Y. Bengio, and H. Lipson, 2014, How transferable are features in deep neural networks?: Advances in neural information processing systems, 3320–3328.

Zadrozny, B., 2004, Learning and evaluating classifiers under sample selection bias: Proceedings of the twenty-first international conference on Machine learning, ACM, 114.